MONTE CARLO CALCULATION OF THE BORN-OPPENHEIMER POTENTIAL BETWEEN TWO HELIUM ATOMS

BY

REX EVERETT LOWTHER

A DISSERTATION PRESENTED TO THE GRADUATE COUNCIL OF
THE UNIVERSITY OF FLORIDA
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE
DEGREE OF DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

1980

ACKNOWLEDGMENTS

I would like to express my deep gratitude to Dr. R. L. Coldwell who developed the techniques that made this work possible, who worked closely with me in all stages of this work, and who was a constant source of guidance and encouragement. I would like to extend my appreciation to Professor A. A. Broyles for his continuous support and encouragement and whose comments and criticisms were extremely helpful in this work. In addition, I would like to thank both Professor A. A. Broyles and R. L. Coldwell for their careful corrections of the first draft of this dissertation. I would like to thank Ron Schoenau and the NERDC computing center for making available the hours of CPU time necessary for the completion of this work. I would also like to thank the NERDC operations staff (many of whom I have become friends with) for service far beyond the call of duty. Finally, I thank my wife, Joni, for the use of her electrons during the preparation of this manuscript.

TABLE OF CONTENTS

Abstract of Dissertation Presented to the
Graduate Council of the University of Florida
in Partial Fulfillment of the Requirements for the
Degree of Doctor of Philosophy

MONTE CARLO CALCULATION OF THE
BORN-OPPENHEIMER POTENTIAL BETWEEN
TWO HELIUM ATOMS

By

Rex Everett Lowther

December, 1980

Chairman: Dr. Arthur A. Broyles
Major Department: Physics

The results of the calculations of extremely accurate wave functions for the ground state of two helium atoms, including energies obtained from these wave functions, are presented herein. These energies provide a variational upper bound to the Born-Oppenheimer potential curve for this system. The necessary expectation values were calculated by biased Monte Carlo techniques at seven internuclear distances. The energy obtained from the trial wave function at the potential minimum is -11.6149685 ± 0.0000030 Ry giving a well depth of $-7.10 \pm 0.30 \times 10^{-5}$ Ry at the nuclear separation distance of 5.6 Bohr radii (a_B). It is estimated that this energy is above the energy of the exact wave function by no more than 1.8×10^{-6} Ry. The extremely small Monte Carlo standard deviation (σ) of 3.0×10^{-6} Ry was made possible through a combination of the three factors:

1. Evaluation of the integrands for many (over 10^6) Monte Carlo points. For the seven internuclear distances this took a total of about 50 hours of CPU time on an Amdahl 470/V6.
2. Monte Carlo methods (which allowed for analytic removal of all singularities) for finding good weight function.
3. The extremely accurate wave functions reported herein.

These wave functions, in fact, were found by minimizing, rather than the energy (<H>), the standard deviation in this energy (σ) which is zero for a perfect wave function. This enabled us to optimize the set of values for the 29 variational parameters by using very few Monte Carlo points and, therefore, made this step financially feasible.

Monte Carlo evaluation of the integrals allows total freedom to choose a natural and concise expansion for the wave functions. The wave functions used combine Schwartz's 189-term Hylleraas-type atomic wave function with molecular terms containing dipole-dipole, dipole-quadrupole, and further terms in the expansion of the interatomic potential energy.

The Born-Oppenheimer potential curve found in this work is in rough agreement with the experimental results of Burgmans, Farrar, and Lee (BFL). The greatest departure is at the nuclear separation distance of 5.6 a_B where the potential found is 1.3σ below the BFL result of -6.70 Ry.

Therefore the upper bound found herein should be considered to be in agreement with the BFL potential curve with just a hint that the exact curve is deeper than the BFL curve.

CHAPTER I
INTRODUCTION

History

The importance and extremely small size of the well depth has made the ground-state helium-helium pair potential the subject of much theoretical and experimental attention with estimates of this depth ranging from -9.1 to -13.5 K. The recent experimental curves begin with Bruch and McGee,[1] who in 1970 fitted a pair potential with a well depth of -10.75 K to dilute-gas properties. Also in 1970, Bennewitz et al.[2] found a well depth of -10.4 K from total scattering cross-section measurements. Differential scattering cross-section measurements were made in 1973 by Farrar and Lee,[3] who found a well depth of -11.0 ± 0.2 K. Burgmans, Farrar, and Lee[4] (hereafter BFL) in 1976 revised this experiment, obtaining a depth of -10.57 K. Nuclear-spin relaxation in dilute gases has also been recognized as a sensitive means of studying intermolecular forces. Chapman[5] in 1975 performed measurements of this on dilute helium gas, finding a potential of the Bruch-McGee form but with a deeper well depth of -11.5 K.

The theoretical work begins in 1931 with a paper by Slater and Kirkwood,[6] who found a helium-helium potential (depth = 9.1 K) by joining a repulsive energy term, which

worked well for small internuclear separations, with an attractive dipole-dipole interaction which they could calculate at large distances. Margenau,[7] also in 1931, extended this formalism to include dipole-quadrupole and quadrupole-quadrupole interactions. This lowered the curve to a depth of -13.5 ± 1.5 K, with the quadrupole-quadrupole term accounting for only 3% of the depth at the minimum. Configuration-interaction (CI) calculations were carried out in 1970-1972 by McLaughlin and Schäfer[8] (-12.0 K), Bertoncini and Wahl[9] (-12.0 K), and others. Attempts to correct or account for the neglect of a large (200-2000 times the size of the well depth) part of the intra-atomic correlation energy missed by these calculations were done by Liu and McLean[10] (-11.0 K), Bertoncini and Wahl[11] (-10.8 K), Dacre[12] (-10.54 K), and Burton[13] (-10.55 K). A good discussion of the problem this correlation energy poses to CI calculations is given in Ref. 10. A calculation (1976) using perturbation theory was carried out by Chalasinski and Jeziorski,[14] who found a lower "bound" of -13.4 K. When they approximately corrected for intra-atomic correlation effects, an upper "bound" of -10.7 K was obtained.

Synopsis

At the beginning of this work, innovative biased selection Monte Carlo techniques[15-17] (developed by R. L. Coldwell in this department) were available for solving the

Schrodinger equation. Coldwell and I felt that these methods are often more precise by orders of magnitude than other presently used Monte Carlo (MC) methods.[18-20] We also felt that these techniques could compete favorably with other existing procedures of evaluating molecular properties such as the configuration integral (CI) method.[8-13,21] We therefore set out to find (successfully) a molecular system that, when solved, would best show:

1. The accuracy that is attainable with these new techniques.
2. The advantages of the MC method over the CI method and other procedures.

We chose the ground state of the di-helium system because it is a system to which a variety of techniques have been applied by many groups[1-14] and because the extremely small well depth required a very high accuracy (1 part in 10^6). Furthermore, with the existence of extremely accurate atomic wave functions,[22-25] and an appealing notion of how to construct molecular wave functions from them,[6-7] the required accuracy seemed attainable.

The variational method most often used[8-13,21] (CI methods) to solve the Schrodinger equation for electronic systems involves the diagonalization of large N x N matrices for wave functions of the form

$$\Psi(\tilde{x}) = \sum_{i}^{N} c_i \phi_i(\tilde{x}) \qquad (1\text{-}1)$$

to find the set of N coefficients $\{c_i\}$ that minimizes the energy. One problem with this method is that the choice of ϕ_i's is limited to a set for which the integrals,

$$N_{ij} = \int \phi_i(\tilde{x}) \phi_j(\tilde{x}) d\tilde{x} , \qquad (1\text{-}2)$$

$$H_{ij} = \int \phi_i(\tilde{x}) H \phi_j(\tilde{x}) d\tilde{x} ,$$

can be evaluated analytically. Since these ϕ's are either Gaussians or atomic-like orbitals, the expansion of the wave function for systems of more than one nucleus is not a natural one. Thus it becomes necessary to make N so large that the matrix manipulations become difficult. For the di-helium system, the accuracy requirements are so great that the energies found by this method are higher than the true energies by amounts 200-2000 times the size of the well depth. The depth is then determined by differencing the energies from that of the infinitely separated atoms based on the assumption that errors in the energy cancel.

If the integrals are evaluated by Monte Carlo, the aforementioned restrictions no longer apply. This allowed us to choose an atomic wave function[25] that included essentially all (to within 5.0×10^{-9} Ry) of the intra-atomic correlation energy. Since differencing from the infinite nuclear separation distance is therefore unnecessary, the energies calculated remain variational upper bounds. These atomic wave functions were then appropriately combined with a natural expansion for the molecular terms containing 29

variational parameters. These terms include dipole-dipole, dipole-quadrupole, quadrupole-quadrupole terms, a term that comes close to including all the terms in the above expansion, and a direct electron-electron interaction term for electrons "on" different nuclei. The exact form for the wave functions and all the constants necessary to duplicate these wave functions are reported herein.

Since the parameters do not appear linearly in our form for the wave functions, as in Eq. (1-1), they had to be optimized by a method other than by a single matrix inversion. Furthermore, due to the Monte Carlo uncertainty of the energy, optimization by minimizing the energy is quite difficult. However, as the perfect wave function is approached, not only does the energy minimize, but the dispersion in this energy becomes zero. Minimizing the Monte Carlo standard deviation, σ, of the energy (which is very similar to this dispersion) over a limited number of MC points is relatively easy because it is a sum of squares. This allowed us to use a much smaller set of MC points (1000) to find the wave functions than were necessary (10^{-5}-10^{-6}) to find the energies of these wave functions. The simplex method[26] was then used to find the parameter set with the lowest σ. The simplex method is slow and sometimes unreliable at finding the absolute minimum; but, since there were only 29 parameters to find, it still consumed much less computer time than is needed to get the same result by diagonalizing the large matrices of the CI method.

Monte Carlo calculations obey the relation

$$\sigma = c/\sqrt{N} \qquad (1\text{-}3)$$

for large N where N is the number of evaluations of the integrand (this ranged from 377,000 to 1159,000 in the final energy estimates) and c is a proportionality constant. Trying to make this standard deviation small by increasing N is feasible only to a certain point, however. Beyond this point the effort is better spent at making the proportionality constant, c, small. We made c small by a combination of the accurate wave functions mentioned above and the biased selection techniques used.

The biased selection method chooses each MC point with a probability proportional to a weight function, then divides by this weight function to correct (rigorously) for the bias in picking it. To produce a small c, the weight function is adjusted to sample most heavily the regions where the integrand and variations in the integrand are large. This work uses recent improvements in the MC method that can choose MC points totally uncorrelated from the last point and with more flexible weight functions than used previously.[18-20]

Total energies (representing variational upper bounds to the true energies) and kinetic energies were found for the wave functions at the internuclear separation distances of 4.5, 5.0, 5.6, 6.6, 7.5, 9.0, 15.0 a_B. At the minimum

of the Born-Oppenheimer potential the energy found was -7.10 ± 0.30 Ry (or -11.20 ± 0.47 K) at the N-N distance of 5.6 a_B. This result is 1.3 standard deviations below the experimental result of -6.70 Ry obtained by Burgmans, Farrar, and Lee[4] (BFL).

CHAPTER II
BIASED SELECTION MONTE CARLO

General Methods

The biased selection method[15-17] is used here to estimate multidimensional integrals. The essence of the method is that the multidimensional points $\tilde{x}_i$ are selected with relative probability $w(\tilde{x}_i)$ and then corrected for this bias so that the integrals are given as

$$I = \int f(\tilde{x})d\tilde{x} = \lim_{N\to\infty} I_N \tag{2-1}$$

where

$$I_N = \frac{1}{N} \sum_{i=1}^{N} \frac{f(\tilde{x}_i)}{w(\tilde{x}_i)} . \tag{2-2}$$

In its simplest form this is importance sampling.[27] In one dimension it amounts to introducing a positive definite function $h(x)$ and the variable change:

$$y = \int_0^x h(z)dz / \int h(z)dz, \tag{2-3}$$

so that

$$\int f(x)dx = \int h(z)dz \int_0^1 \frac{f[x(y)]}{h[x(y)]} dy. \tag{2-4}$$

A Monte Carlo evaluation of this integral involves choosing y_i at random, solving Eq. (2-3) for $x_i = x(y_i)$, and finding

$$w(x_i) = h(x_i) / \int h(z)dz. \qquad (2\text{-}5)$$

This is repeated N times and the value of the integral is found from Eq. (2-2). The probability of selecting a given point x_i is proportional to $h(x_i)$. If $h(x)$ is large where $f(x)$ is large and small where it is small, points will be concentrated in the important region of f. In addition, $f(x_i)/w(x_i)$ will tend to be constant which will reduce the Monte Carlo error. Notice that if, in Eq. (2-4), $h[x(y)]$ was made equal to $f[x(y)]$, the dispersion in the integrand would disappear. This can never be practical, though, since that leaves the integral over $h(z)$ in Eq. (2-3) identical to the original integral. Nevertheless, it is often possible to choose a weight function for which $\int h(z)dz$ is easy to evaluate and which matches the function $f(x)$ closely so that the standard deviation is decreased by several orders of magnitude.

A good example is the radial weight function used to pick an electron position $\vec{r}$ with respect to a nucleus. This weight function, which is similar to the square of the Hartree Fock atomic wave function $\phi^2_{HF}(|\vec{r}|)$, closely fits $\Psi^2(\tilde{x})$ in this region of the 3-dimensional subspace of $\tilde{x}$. And yet, since $\phi^2_{HF}(|\vec{r}|)$ is 1-dimensional, $h(\vec{r})$ and its integral are easy to calculate. We do this by evaluating our best choice for the weight function at 65 values of $r = |\vec{r}|$, and setting up a table (Table 1) of r_m, $\hat{h}(r_m) = r_m^2 h(r_m)$, and $I_m = \int_0^{r_m} \hat{h}(r)dr$. The function $\hat{h}(r)$ is

Table 1. The values of r_m, $\hat{h}(r_m)$, and I_m used to pick electron positions with respect to the nuclei.

r_m	$\hat{h}(r_m)$	I_m
0.15	5.13553E-2	7.70329E-3
0.30	5.13553E-2	1.54066E-2
0.45	6.71723E-2	2.42962E-2
0.60	7.09341E-2	3.46541E-2
0.75	6.71994E-2	4.50142E-2
0.90	5.98367E-2	5.45419E-2
1.05	5.13377E-2	6.28799E-2
1.20	4.30723E-2	6.99607E-2
1.35	3.56702E-2	7.58664E-2
1.50	2.93291E-2	8.07413E-2
1.65	2.40218E-2	8.47426E-2
1.80	1.96240E-2	8.80161E-2
1.95	1.59859E-2	9.06868E-2
2.10	1.29678E-2	9.28583E-2
2.25	1.04546E-2	9.46150E-2
2.40	8.35737E-3	9.60259E-2
2.55	6.60954E-3	9.71484E-2
2.70	5.16083E-3	9.80312E-2
2.85	3.97155E-3	9.87161E-2
3.00	3.00809E-3	9.92396E-2
3.15	2.99529E-3	9.96899E-2
3.30	3.07932E-3	1.00145E-1
3.45	3.11867E-3	1.00610E-1
3.60	3.01786E-3	1.01071E-1
3.75	2.87206E-3	1.01512E-1
3.90	2.73679E-3	1.01933E-1
4.05	2.61130E-3	1.02334E-1
4.20	2.49487E-3	1.02717E-1
4.35	2.38686E-3	1.03083E-1
4.50	2.28665E-3	1.03434E-1
4.65	2.19368E-3	1.03770E-1
4.80	2.10743E-3	1.04092E-1
4.95	2.02741E-3	1.04402E-1
5.10	1.95317E-3	1.04701E-1
5.25	1.88430E-3	1.04989E-1
5.40	1.82040E-3	1.05267E-1
5.55	1.76112E-3	1.05535E-1
5.70	1.70613E-3	1.05795E-1
5.85	1.65510E-3	1.06047E-1
6.00	1.60777E-3	1.06292E-1
6.15	1.56385E-3	1.06530E-1
6.30	1.52311E-3	1.06761E-1
6.45	1.48531E-3	1.06987E-1
6.60	1.45025E-3	1.07207E-1
6.75	1.41771E-3	1.07422E-1
6.90	1.38753E-3	1.07633E-1
7.05	1.35953E-3	1.07839E-1
7.20	1.33355E-3	1.08041E-1
7.35	1.30945E-3	1.08239E-1
7.50	1.28709E-3	1.08434E-1
7.65	1.26635E-3	1.08625E-1
7.80	1.24710E-3	1.08814E-1
7.95	1.22925E-3	1.08999E-1
8.10	1.21268E-3	1.09183E-1
8.25	1.19731E-3	1.09363E-1
8.40	1.18306E-3	1.09542E-1
8.55	1.16983E-3	1.09718E-1
8.70	1.15756E-3	1.09893E-1
8.85	1.14617E-3	1.10066E-1
9.00	1.13561E-3	1.10237E-1
9.15	1.12581E-3	1.10406E-1
9.30	1.11672E-3	1.10575E-1
9.45	1.10829E-3	1.10741E-1
9.60	1.10046E-3	1.10907E-1

then defined as the linearly interpolated values between the h_m. Finally, an electron position is picked by the recipe:

1. Choose a random number, ⊠, between zero and I_{65},
2. Determine m such that $I_m < ⊠ < I_{m+1}$,
3. The radial distance is determined from the interpolation:

$$r = r_m + \frac{(⊠ - I_m)}{\hat{h}_m + [\hat{h}_m^2 + \frac{\hat{h}_{m+1} - h_m}{r_{m+1} - r_m} (⊠ - I_m)]^{1/2}} \quad . \tag{2-6}$$

For the purpose of using more than one weight function, one would like a proof of Eq. (2-4) for which no change of variables is needed. Imagine a lattice of T different values of $\tilde{x}_i$, each of which contribute equally to the integral and which has the probability $w(\tilde{x}_i)$ of being selected. It must be shown that

$$\frac{1}{T} \sum_{i=1}^{T}{}' f(\tilde{x}_i) = \lim_{N\to\infty} \frac{1}{N} \sum_{i=1}^{N} \frac{f(\tilde{x}_i)}{w(\tilde{x}_i)} \quad . \tag{2-7}$$

The prime indicates a sum over each of the T points, while the unprimed sum contains the same point many times. In a set of points large compared to T, the point $\tilde{x}_i$ will be found $Nw(\tilde{x}_i)/T$ times since, by assumption, its relative probability of being selected is $w(\tilde{x}_i)$. Grouping the same values of $\tilde{x}_i$ together, there are $Nw(\tilde{x}_i)/T$ terms of value $f(\tilde{x}_i)/w(\tilde{x}_i)$. The relative probability cancels to leave a

total value for each point $\tilde{x}_i$ of $Nf(\tilde{x}_i)/T$. This is exactly the value needed to give Eq. (2-7) explicitly. Taking the limit $T \to \infty$ so that all values of $\tilde{x}_i$ are possible yields Eq. (2-1). The positions $\tilde{x}_i$ may be picked with any relative probability $w(\tilde{x}_i)$ as long as it is divided out in Eq. (2-2).

One last step is needed to arrive at the MC method as used here. Suppose that there are two independent methods for choosing points and that method 1 is used with probability p, and method 2 with probability $p_2 = 1 - p_1$. The N points will contain $Np_1w_1(\tilde{x}_i)/T$ terms picked with method 1. If both $w_1(\tilde{x}_i)$ and $w_2(\tilde{x}_i)$ in Eq. (2-2) are replaced by the overall probability of finding $\tilde{x}_i$,

$$w(\tilde{x}_i) = p_1w_1(\tilde{x}_i) + p_2w_2(\tilde{x}_i), \tag{2-8}$$

we will have terms of total value

$$[Np_1w_1(\tilde{x}_i)/T][f(\tilde{x}_i)/w(\tilde{x}_i)] + [Np_2w_2(\tilde{x}_i)/T][f(\tilde{x}_i)/w(\tilde{x}_i)]$$

$$= Nf(\tilde{x}_i)/T \tag{2-9}$$

which is exactly the value needed for Eq. (2-7).

Removal of Singularities

This last result allows us to choose points with different probability distributions and average over the probabilities of getting final points. This made it possible to

find the weight function needed to produce small standard deviations and to remove the singularities from $H\Psi(\tilde{x})$.

For example, suppose we want the three dimensional integral in the region $|\vec{r}| < 10$ of the function

$$f(\vec{r}) = 1/|\vec{r}-\vec{a}| + 1/|\vec{r}-\vec{b}| + g(\vec{r}) \qquad (2\text{-}10)$$

where $g(\vec{r})$ has no singularities. The singularities would normally introduce large dispersions owing to the fact that points are eventually picked arbitrarily close to $\vec{a}$ and $\vec{b}$ which requires an enormous number of points to average down. If, however, we introduce the weight functions

$$\begin{aligned} w_1(\vec{r}) &= c_1/|\vec{r}-\vec{a}|^2, \quad |\vec{r}-\vec{a}| < 1 \\ &\quad 0, \qquad\qquad |\vec{r}-\vec{a}| > 1 \\ w_2(\vec{r}) &= c_2/|\vec{r}-\vec{b}|^2, \quad |\vec{r}-\vec{b}| < 1 \\ &\quad 0, \qquad\qquad |\vec{r}-\vec{b}| > 1 \\ w_3(\vec{r}) &= c_3, \qquad\qquad |\vec{r}| < 10 \\ &\quad 0, \qquad\qquad |\vec{r}| > 10 \end{aligned} \qquad (2\text{-}11)$$

with the probabilities $p_1 = p_2 = p_3 = 1/3$, we are then required to pick these w's which choose $\vec{r}$ with equal probability. (To prevent ever picking any points outside the region of integration, assume $|\vec{a}|, |\vec{b}| \leq 9.0$.) The constants c_i are determined by the condition

$$\int w(\vec{r})d\vec{r} = 1 \ . \qquad (2\text{-}12)$$

We find then the average weight function given by

$$w(\vec{r}) = \frac{1}{2\pi}\left(\frac{1}{|\vec{r}-\vec{a}|^2} + \frac{1}{|\vec{r}-\vec{b}|^2} + 1.5 \times 10^{-3}\right) \qquad (2\text{-}13)$$

which goes to infinity faster than $f(\vec{r})$ for $\vec{r}$ close to $\vec{a}$ or $\vec{b}$. It can be seen from Eq. (2-2) that this eliminates the singularities.

Suppose now that these weight functions are not averaged. We would then get, instead of Eq. (2-9), terms of total value

$$w(\vec{r}_i) = [Np_1 w_1(\vec{r}_i)/T][f(\vec{r}_i)/w_1(\vec{r}_i)] + [Np_2 w_2(\vec{r}_i)/T][f(\vec{r}_i)/w_2(\vec{r}_i)] + \ldots . \qquad (2\text{-}14)$$

However, it is still possible to pick points arbitrarily close to $\vec{a}$ or $\vec{b}$ with the weight function $w_3(\vec{r}_i)$. These terms, since they are divided by $w_3(\vec{r}_i)$ rather than $w(\vec{r}_i)$, still have singularities. Terms like this with large values of $f(\tilde{x})$ but small values of $w(\tilde{x})$ (producing high standard deviations) are eliminated by forming a $w(\tilde{x}_i)$ averaged over all possible ways of finding $\tilde{x}_i$. By taking advantage of symmetries in $f(\tilde{x})$, we were even able to average over some values of $\tilde{x}$ that produced values identical to $f(\tilde{x}_i)$.

A Monte Carlo Method not Used Herein

In the biased selection method explained above, each MC point $\tilde{x}_i$ is picked totally independently from all other points. There is another method--the Markov chain

method[18]--which picks the next point $\tilde{x}_{i+1}$ randomly within a small box centered on the last point $\tilde{x}_i$. This move is then accepted or rejected according to a weight function $w(\tilde{x})$:

If $w(\tilde{x}_{i+1}) \geq w(\tilde{x}_i)$, the move is accepted.

If $w(\tilde{x}_{i+1}) < w(\tilde{x}_i)$, the move is accepted with probability $w(\tilde{x}_{i+1})/w(\tilde{x}_i)$.

When the move is rejected, $\tilde{x}_{i+1}$ is set equal to $\tilde{x}_i$. It can be shown that this weight function corresponds exactly to the weight function used in Eq. (2-2). This proof depends upon the ergodicity condition that any position $\tilde{x}$ can eventually be reached from any other position.

By choosing $w(\tilde{x})$ equal to the weight function used in the biased selection method, it is possible for the Markov chain to achieve the same results. The singularities in $w(\tilde{x})$, however, can cause troubles for the Markov chain. Suppose, as in the example above, that $\vec{r}_i$ is picked close to point $\vec{a}$ such that $|\vec{r}_i - \vec{a}| = 0.01$. A new location picked farther from the singularity, for example $|\vec{r}_{i+1} - \vec{a}| = 0.10$, will have a probability

$$P = w(\vec{r}_{i+1})/w(\vec{r}_i) = (c_1/.10)^2/(c_1/.01)^2 = 0.01 \qquad (2\text{-}15)$$

of being accepted. This shows that by repeatedly rejecting points farther away from $\vec{a}$ than $\vec{r}_i$, the Markov chain can become "stuck" in these regions near the singularities. Remedying this problem is possible, but introduces complexities. Most have avoided it by leaving out the

singular terms in w. The singularities in f must then be handled by other methods. The problem is avoided altogether in this work because an $\tilde{x}_i$ chosen by the biased selection method has no bearing on the position of the next point $\tilde{x}_{i+1}$.

Standard Deviation

The Monte Carlo standard deviation in the value found in Eq. (2-2) can be found by squaring the equation

$$I-I_N = \sum_{i=1}^{N} \delta_i \quad . \tag{2-16}$$

and taking an ensemble average (where δ_i is the error introduced into I_N by omitting the i'th point):

$$\sigma_N^2 = \langle (I-I_N)^2 \rangle = \left\langle \sum_{i=1}^{N} \sum_{j=1}^{N} \delta_i \delta_j \right\rangle = \left\langle \sum_{j=1}^{N} \delta_j^2 \right\rangle$$

$$\approx \sum_{j=1}^{N} \delta_j^2 \quad . \tag{2-17}$$

The nondiagonal elements cancel over an ensemble average because each MC point is totally uncorrelated with any previously picked point. The quantity $\Sigma_j \delta_j^2$, because it is a sum of squares, is very close to its ensemble average. In practice we found that, for 10 runs with N = 2000, this quantity varied by around 2% and by 1% for N = 10,000. The standard deviation in I_N is then

$$\sigma_N^2 = \sum_{j=1}^{N} \delta_j^2$$

$$= \sum_{j=1}^{N} \left(\frac{1}{N} \sum_{i=1}^{N} \frac{f(\tilde{x}_i)}{w(\tilde{x}_i)} - \frac{1}{N-1} \sum_{i=1}^{N} \frac{f(\tilde{x}_i)}{w(\tilde{x}_i)} + \frac{1}{N-1} \frac{f(\tilde{x}_j)}{w(\tilde{x}_j)} \right)^2$$

$$= \sum_{j=1}^{N} \left(\frac{1}{N(N-1)} \sum_{i=1}^{N} \frac{f(\tilde{x}_i)}{w(\tilde{x}_i)} + \frac{1}{N-1} \frac{f(\tilde{x}_j)}{w(\tilde{x}_j)} \right)^2$$

$$= \sum_{j=1}^{N} \frac{-1}{(N-1)^2} \left(\frac{f(\tilde{x}_i)}{w(\tilde{x}_i)} \right)^2 - \frac{1}{N(N-1)^2} \left(\sum_{i=1}^{N} \frac{f(\tilde{x}_i)}{w(\tilde{x}_i)} \right)^2$$

$$\approx \frac{1}{N} \left[\frac{1}{N} \sum_{i=1}^{N} \left(\frac{f(\tilde{x}_i)}{w(\tilde{x}_i)} \right)^2 - \left(\frac{1}{N} \sum_{i=1}^{N} \frac{f(\tilde{x}_i)}{w(\tilde{x}_i)} \right)^2 \right] \qquad (2\text{-}18)$$

for large N. The dependence of σ^2 on N is readily seen as $\sigma \propto 1/\sqrt{N}$.

Calculating the Energy

The energy of a trial wave function $\psi(\tilde{x})$ is

$$E = \langle H \rangle = \frac{\int \psi(\tilde{x}) H \psi(\tilde{x}) d\tilde{x}}{\int \psi^2(\tilde{x}) d\tilde{x}} . \qquad (2\text{-}19)$$

Applying Eq. (2-2) to both integrals gives

$$E_N = \frac{\sum_{i=1}^{N} \psi(\tilde{x}_i) H \psi(\tilde{x}_i) / w(\tilde{x}_i)}{\sum_{i=1}^{N} \psi^2(\tilde{x}_i) / w(\tilde{x}_i)} . \qquad (2\text{-}20)$$

When summed over the same set of points $\tilde{x}_i$, the numerator and denominator are highly correlated. It can be seen from Eq. (2-20) that this eliminates the part of the dispersion

resulting from variations in $\Psi^2(\tilde{x})/w(\tilde{x})$, and leaves only contributions from variations in $H\Psi(\tilde{x})/\Psi(\tilde{x})$ [weighted by $\psi^2(\tilde{x})/w(\tilde{x})$]. Since $H\Psi(\tilde{x})/\Psi(\tilde{x})$ becomes constant for the correct wave function, the dispersion of E_N in Eq. (2-20) will be much smaller than the dispersions of either the numerator or denominator. This standard deviation in E_N can be calculated by the same procedure leading to Eq. (2-18):

$$\sigma_N^2 = \sum_{j=1}^{N} \delta_j^2 = \sum_{j=1}^{N} \left(\frac{\sum_{i=1}^{N} E_i \Psi^2(\tilde{x}_i)/w_i}{\sum_{i=1}^{N} \Psi^2(\tilde{x}_i)/w_i} - \frac{\sum_{i=1}^{N} E_i \Psi^2(\tilde{x}_i)/w_i - E_j \Psi^2(\tilde{x}_j)/w_i}{\sum_{i=1}^{N} \Psi^2(\tilde{x}_i)/w_i - \Psi^2(\tilde{x}_j)/w_i} \right)^2 \qquad (2\text{-}21)$$

where $E_i = H\Psi(\tilde{x}_i)/\Psi(\tilde{x}_i)$ is the local energy, and $w_i = w(\tilde{x}_i)$. Expanding the second term to first order gives

$$\sigma_N^2 = \sum_{j=1}^{N} \left(\frac{E_j \Psi^2(\tilde{x}_j)/w_j}{\sum_{i=1}^{N} \Psi^2(\tilde{x}_i)/w_i} - \frac{\sum_{i=1}^{N} E_i \Psi^2(\tilde{x}_i)/w_i}{\sum_{i=1}^{N} \Psi^2(\tilde{x}_i)/w_i} \frac{\Psi^2(\tilde{x}_j)/w_j}{\sum_{i=1}^{N} \Psi^2(\tilde{x}_i)/w_i} \right)^2$$

$$= \sum_{j=1}^{N} (E_j \Psi^2(\tilde{x}_j)/w_j - E_N \Psi^2(\tilde{x}_j)/w_j)^2 / \left(\sum_{i=1}^{N} \Psi^2(\tilde{x}_i)/w_i \right)^2 . \qquad (2\text{-}22)$$

It can be seen from this expression that when $\Psi(\tilde{x})$ is an eigenfunction of H then the standard deviation is zero. This was, in fact, the quantity minimized (see Ch. III) to find wave functions as close as possible to the correct eigenfunctions.

Since the quantity we now want to calculate is a ratio of two integrals, the criterion for choosing the optimum $w(\tilde{x})$ is slightly different. In fact, it is now impossible to eliminate the MC dispersion by choosing a weight function that perfectly matches the integrand (since there are two integrands). However, from this criterion alone, one can construct a pretty good weight function by coming close to the relation

$$w(\tilde{x}) \propto \Psi^2(\tilde{x}) . \tag{2-23}$$

This is the weight function most often used with the Markov chain method.[18-20] Based on the assumption that variations of $H\Psi(\tilde{x})/\Psi(\tilde{x})$ do not change much from region to region, this is also the best $w(\tilde{x})$.

We have, however, found repeatedly that these variations become greater as $\psi^2(\tilde{x})$ becomes small. This is because it is always most efficient to optimize wave functions with greatest emphasis in the regions where $\Psi^2(\tilde{x})$ is large. This is especially true since there is a much greater volume in $\tilde{x}$-space where $\Psi^2(\tilde{x})$ is small than where it is large. Consequently, it is harder to fit $\Psi^2(\tilde{x})$ to this vast region with a limited set of variational

parameters. Therefore, we found that it is profitable to choose a weight function that selects more points [than the weight function in Eq. (2-23)] into the regions where $\Psi^2(\tilde{x})$ is small.

Consider the one-dimensional example over a region where $\Psi^2(x)$ is constant and $H\Psi(x)/\Psi(x)$ is given by Fig. 1. Out of 100 MC points, a constant weight function obeying the relation $w(x) = \Psi^2(x) = 1$ will pick on the average 90 points in region 1 and 10 points in region 2. There will be 45 points each for which $H\Psi(x)/\Psi(x)$ equals 21 and 19, and 5 points each for the values 10 and 30. Substituting this into Eq. (2-22) gives

$$\sigma^2 = [45(21-20)^2 + 45(19-20)^2 + 5(30-20)^2 + 5(10-20)^2]/(90+10)^2$$

$$= (90+1000) \times 10^{-4} = .109 . \qquad (2-24)$$

Notice that over 90% of the contribution to σ^2 is coming from region 2 (R2). If we now choose a weight function that quadruples the number of points in region 2 at the expense of R1, we have from Eq. (2-8)

$$w(x) = 2/3, \ x \text{ in R1} \qquad (2-25)$$

$$= 4 \ , \ x \text{ in R2}$$

and

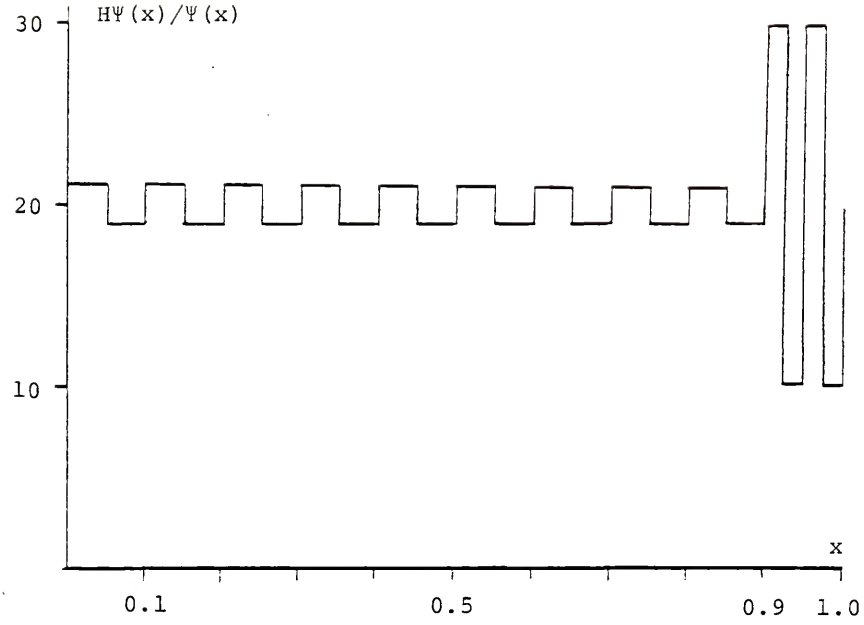

Figure 1. Values of $H\Psi(x)/\Psi(x)$ for an example one-dimensional trial wave function. Positions between 0.0 and 0.9 are in region 1, and in region 2 if between 0.9 and 1.0.

$$\sigma^2 = [30(\frac{21-20}{2/3})^2 + 30(\frac{19-20}{2/3})^2 + 20(\frac{30-20}{4})^2$$

$$+ 20(\frac{10-20}{4})^2]/[60\,\frac{1}{2/3} + 40\,\frac{1}{4}]^2$$

$$= (135 + 250)\times 10^{-4} = 0.0385 . \qquad (2\text{-}26)$$

The contribution to σ^2 from R1 is only slightly increased while the contribution from R2 is cut by 75%. By throwing extra MC points into the region where the oscillations in $H\Psi(x)/\Psi(x)$ are large, we have found a weight function significantly better than $w(x) = \Psi^2(x)$.

The weight function we used for the di-helium system was a compromise between these two considerations. About 1/4 of our MC points were picked in the region where $\Psi^2(\tilde{x})$ is large; the rest of the points were picked in regions where [although $\Psi^2(\tilde{x})$ is small] oscillations in $H\Psi(\tilde{x})/\Psi(\tilde{x})$ were expected to be large. Although the contribution to σ^2 from the region where $\Psi^2(\tilde{x})$ is large is higher than if $w(\tilde{x}) = \Psi^2(\tilde{x})$, the overall standard deviation is reduced significantly. We also gain assurance that all regions of $\tilde{x}$-space are sampled adequately.

A good measure of the number of points that are chosen in the regions where $\Psi^2(\tilde{x})$ is large is given by the quantity we call the effective N:

$$N_e = \frac{[\sum_{i=1}^{N} \Psi^2(\tilde{x}_i)/w_i]^2}{\sum_{i=1}^{N} \Psi^4(\tilde{x}_i)/w_i^2} . \qquad (2\text{-}27)$$

If the points were chosen totally at random, one large value would dominate both sums resulting in $N_e = 1$. If the weight function in Eq. (2-23) was used, all points would be equal in both sums giving $N_e = N$. As implied above, we found that N_e was about 1/4 of the total number of points N.

Weight Functions

In this work, an electron position $\vec{r}_k$ in the point $\tilde{x}$ had probabilities of being picked in the different ways:

1. With respect to nucleus A (or B). In this case the distance $r_{kA} = |\vec{r}_{kA}|$ was chosen with a constant probability for $r_{kA} < 0.30\ a_B$ [introducing a $1/r_{kA}^2$ into w_1 (or w_2)] and for larger distances with probability r_{kA}^2 times the square of the Hartree-Fock (HF) atomic wave function [introducing the square of the HF atomic wave function into w_1 (or w_2)]. A slight alteration was made in the HF wave functions to throw extra points (about 5%) into regions where r_{kA} [and variations in $H\Psi(\tilde{x})/\Psi(\tilde{x})$] is large.
2. With respect to a point M midway between the nuclei. In this case the distance r_{kM} was chosen with probability r_{kM} (introducing a $1/r_{kM}$ into w_3). The maximum r_{kM} that could be picked with this weight function (w_3) was one half the internuclear separation, R_{AB}.

3. With respect to a large box (40 x 40 x 60 a_B) centered on the nuclei. In this case the positions $\vec{r}_k$ were selected randomly within the box. The weight function w_4 is then just a constant determined by the normalization condition described above.

4. With respect to any previously picked electron $\vec{r}_j$. In this case the distance $r_{kj} = |\vec{r}_k - \vec{r}_j|$ was chosen with constant probability up to a maximum distance of 1.3 a_B. These weight functions (w_5-w_8) are then proportional to $1/r_{kj}$.

The above weight functions were normalized in the manner of Eq. (2-5) and Eq. (2-12) such that (see Table 2)

$$\int w_i(\vec{r})d\vec{r} = \int w_j(\vec{r})d\vec{r} \,. \tag{2-28}$$

An overall normalization factor is unnecessary since the factor $\int h(\tilde{x})d\tilde{x}$ appears in both numerator and denominator in Eq. (2-20).

The probabilities ${}^k p_i$ of picking an electron with the weight functions $w_i(\vec{r}_k)$ (Table 3) depend on the order in which this electron is picked. The average weight function for the k'th picked electron is then determined by

$${}^k w(\vec{r}_k) = \sum_{i=1}^{8} {}^k p_i w_i(\vec{r}_k) \,. \tag{2-29}$$

If not averaged further, the total weight function then becomes

Table 2. Information relevant to weight functions $w_i(\vec{r}_k)$ used in Eq. (2-28) and Eq. (2-29). ϕ_{HFM} is the modified Hartree-Fock wave function. The constant I is defined by $I = \int_0^{9.6} w_1(r) r^2 dr$ and is calculated numerically within the computer code (as explained on pages 9-11).

i	$\vec{r}_k$ picked with respect to:	domain	form for $w_i(\vec{r}_k)$	normalization constants, c_i
1	nucleus A	$r_{kA} < 9.6\ a_B$	$\phi^2_{HFM}(r_{kA})$, $r_{kA} \geq .3\ a_B$ $\phi^2_{HFM}(.3)\left(\frac{.3}{r_{kA}}\right)^2$, $r_{kA} < .3\ a_B$	---
2	nucleus B	$r_{kB} < 9.6\ a_B$	similar to w above	---
3	box	$\lvert x_k\rvert, \lvert y_k\rvert < 20\ a_B$, $\lvert z_k\rvert < 30\ a_B$	C_3	$\frac{4\pi I}{(40)(40)(60)}$
4	midpoint between A and B	$r_{kM} < R_{AB}/2$	C_4/r_{kM}	$8I/r^2_{AB}$
5	electron 1	$r_{k1} < 1.3\ a_B$	C_5/r^2_{k1}	$I/1.3$
6	electron 2	$r_{k2} < 1.3\ a_B$	C_6/r^2_{k2}	$I/1.3$
7	electron 3	$r_{k3} < 1.3\ a_B$	C_7/r^2_{k3}	$I/1.3$
8	electron 4	$r_{k4} < 1.3\ a_B$	C_8/r^2_{k4}	$I/1.3$

Table 3. Probabilities ${}^{k}p_i$ used in Eq. (2-29) to pick the k'th electron with the weight functions $w_i(\vec{r})$.

${}^{k}p_i$	1st nucleus	2nd nucleus	box	midpoint	previously picked electrons
1st electron	85/110	5/110	3/110	17/110	0
2nd electron	5/107	85/107	3/107	7/107	7/107
3rd electron	85/114	5/114	3/114	7/114	(2x7)/114
4th electron	5/121	85/121	3/121	7/112	(3x7)/121

$$w(\tilde{x}) = {}^{1}w(\vec{r}_1)\,{}^{2}w(\vec{r}_2)\,{}^{3}w(\vec{r}_3)\,{}^{4}w(\vec{r}_4)\ . \qquad (2\text{-}30)$$

However, it is not only possible to average over all weight functions leading to the chosen value of $\tilde{x}$ [see proof leading to Eq. (2-9)], but it is also correct to average over weight functions $w(\tilde{x}')$ with different positions for which the integrands are identical such that $f(\tilde{x}') = f(\tilde{x})$. And, since our wave function is antisymmetric with respect to interchange of parallel-spin electrons, the values $\Psi(\tilde{x})H\Psi(\tilde{x})$ and $\Psi^2(\tilde{x})$ are invariant with respect to these permutations. Thus it is possible to average $w(\tilde{x})$ over these permutations:

$$w(\tilde{x}) = (1 + p_{12} + p_{34} + p_{12}p_{34})\,{}^{1}w(\vec{r}_1)\,{}^{2}w(\vec{r}_2)\,{}^{3}w(\vec{r}_3)\,{}^{4}w(\vec{r}_4) \qquad (2\text{-}31)$$

where the first two electrons picked are labeled spin-up, the last two, spin-down. Since there are no external fields present, the integrands also remain invariant when all electron spins are flipped. This final averaging led to the weight function actually used,

$$w(\tilde{x}) = (1 + p_{13}p_{24})(1 + p_{12} + p_{34} + p_{12}p_{34})$$

$$\qquad {}^{1}w(\vec{r}_1)\,{}^{2}w(\vec{r}_2)\,{}^{3}w(\vec{r}_3)\,{}^{4}w(\vec{r}_4), \qquad (2\text{-}32)$$

and lowered σ by about 3% from the weight function in Eq. (2-31).

CHAPTER III
FINDING THE BEST SET OF VARIATIONAL PARAMETERS

Criterion Used

After deciding on a form for the trial wave function (Ch. IV), the optimum set of m variational parameters {c} must be found. When all coefficients appear linearly in $\Psi(\tilde{x})$, as in the CI method,[21] a single matrix inversion is sufficient to solve the set of m simultaneous equations

$$\frac{\partial}{\partial c_k}\left[\frac{\int \check{\Psi}(\tilde{x}) H \Psi(\tilde{x}) d\tilde{x}}{\int \Psi^2(\tilde{x}) d\tilde{x}}\right] = 0, \quad k = 1,m. \tag{3-1}$$

This produces the lowest energy possible with the form used and finds the best set of coefficients {c} in one step. The terms in $\Psi(\tilde{x})$, however, are limited to a form for which the matrix elements in Eq. (1-2) are very precisely known. This rules out terms that depend directly on inter-electron distances and introduces orthogonality constraints.

Here the linear form is dropped in favor of a more natural expansion of $\Psi(\tilde{x})$ with 29 variational parameters. Our wave functions, then, are such that all integrals must be evaluated by the Monte Carlo method and such that the parameters {c} must be found by other than a single matrix inversion. Also, since the MC method can only find an

estimate to the integral, computational difficulties arise when minimizing the energy.

Suppose we were to minimize the energy by the following method:

1. Select a group of MC points $\{\tilde{x}\}_{min}$.
2. Use a searching routine to find the parameter set $\{c\}$ which gives the lowest MC estimate to the energy over $\{\tilde{x}\}_{min}$.

Suppose further that c_1 is capable of making the local energy, $H\Psi(\tilde{x})/\Psi(\tilde{x})$, at the point $\tilde{x}_i$ become

$$\frac{H\Psi(\tilde{x}_i)}{\Psi(\tilde{x}_i)} = \langle H \rangle + c_1\alpha \qquad (3\text{-}2)$$

(where α is a positive constant) and with no effect on any other point in $\{\tilde{x}\}_{min}$. (Since $\{\tilde{x}\}_{min}$ was restricted to 1000 MC points, they are spread sparsely enough to make this a realistic situation.) The MC estimate to the energy can be made arbitrarily small by making $c_1 \to -\infty$, even though we know that the correct value for c_1 is zero. The true energy gets no lower, of course, because the local energy for unsampled regions of $\tilde{x}$ increases as the few sampled regions go low. By increasing the number of points in $\{\tilde{x}\}_{min}$ so that a small variation in any parameter affects the local energy at many of these points, it is still possible to optimize $\Psi(\tilde{x})$ by minimizing the energy. But, since several hundred sets of variational parameters must

be tested by the optimizing routine, it is too expensive to let $\{\tilde{x}\}_{min}$ become this large.

Fortunately, it is not necessary to optimize the trial wave functions by minimizing the energy. Since all eigenfunctions of the Hamiltonian have zero dispersion in the local energy, we were able to minimize a quantity closely related to this dispersion--the MC standard deviation. [The standard deviation actually used (described more fully in Ch. V) is the uncertainty in the difference of two energies for different nuclear separation distances and is very similar to that defined in Eq. (2-22).] A similar local energy method was used by Frost et al.[28] but without weight functions. This forced them to waste constants making $H\Psi(\tilde{x})/\Psi(\tilde{x}) = \langle H \rangle$ in regions of $\tilde{x}$ where $\Psi^2(\tilde{x})$ is small.

By minimizing σ rather than $\langle H \rangle$ in the above example, the constant c_1 is set to the correct value of zero. This demonstrates that a smaller number of MC points in $\{\tilde{x}\}_{min}$ is sufficient. Since σ^2 is a sum of squares, there is no way to lower this quantity by letting a small number of points dominate over the rest. Consequently, the points must be fitted more equally. This greatly lowers the possible values of $\{c\}$ for which $\Psi(\tilde{x})$ has an artificially low value of σ. The minimization routine is forced to find a wave function close to the eigenfunction in order to get a low σ--even over a relatively small set of points. We used 1000 points in $\{x\}_{min}$ to find the 29 variational

parameters in $\Psi(\tilde{x})$. The obvious price we pay is that the parameter set $\{c\}$ that minimizes σ is slightly different from the $\{c\}$ that gives the lowest possible upper bound to the energy. The wave functions were so accurate, however, that these energies were above the true energies by a very small amount (estimated in Ch. VI).

Selection of $\{\tilde{x}\}_{min}$

Since we are limited to about 1000 points in $\{\tilde{x}\}_{min}$ when optimizing the parameters in $\Psi(\tilde{x})$, it is important to select many of these in the region where $\Psi^2(\tilde{x})$ is large. It is also important to choose some MC points in regions where $H\Psi(\tilde{x})/\Psi(\tilde{x}) \neq \langle H \rangle$. Since this is more frequent when $\Psi^2(\tilde{x})$ is small, the selection of points into the region where $\Psi^2(\tilde{x})$ is large, should not be too efficient. This was carried out by selecting each point in $\{\tilde{x}\}_{min}$ out of a group of 2 with the probability

$$P = \frac{[\Psi^2(\tilde{x}_p)/w(\tilde{x}_p)]^{1/2}}{\sum_{i=1}^{2} [\Psi^2(\tilde{x}_i)/w(\tilde{x}_i)]^{1/2}} \quad . \tag{3-3}$$

This, of course, introduces a new weight function used to calculate σ:

$$w(\tilde{x}_p) \rightarrow Pw(\tilde{x}_p). \tag{3-4}$$

The selected points were then stored, along with the weight functions, for use by the minimization routine in testing the various sets of variational parameters $\{c\}$.

Optimizing Routine

The simplex method[26] was used to find the set of m parameters {c} that minimized σ. This routine first guesses at m+1 sets of parameters, "points," then systematically moves them around in the m-dimensional parameter space until the minimum σ is found. The scheme takes the "highest point" ({c} with highest value of σ) and reflects it in the parameter space about the midpoint of all the other "points." If this point is found to be lower than all the other points, then another point further in this direction is tested; if it is higher than the second highest, then a point closer to the midpoint is selected. The lowest of the tested points then replaces the highest previous point and the process is repeated until a minimum is reached. If none of the points tested above are smaller than the second highest point, then this routine multiplies the distance of each point from the midpoint by a random number (between -b to +b where b varies from 8 to 1/8) before the next iteration.

As with all nonlinear optimization routines, simplex has the two related problems:

1. It is slow. About 100 evaluations of σ per variational parameter are necessary to find the minimum.
2. The routine tends to get stuck in local minima far above global minimum.

Several attempts[29] to speed the convergence of simplex have only increased the second problem. It is important to keep the routine somewhat insensitive to fluctuations in σ so that a wide region of the parameter space is sampled to prevent it from naively converging onto a local minimum.

CHAPTER IV
WAVE FUNCTIONS

General Form

The wave function we used combined extremely accurate atomic wave functions with terms accounting for interactions between the two atoms:

$$\Psi(\vec{r}_1,\vec{r}_2,\vec{r}_3,\vec{r}_4) = (1-P_{12}-P_{34}+P_{12}P_{34}) \times\psi_A(\vec{r}_1,\vec{r}_3)\psi_B(\vec{r}_2,\vec{r}_4)\exp[-U(\vec{r}_1,\vec{r}_3;\vec{r}_2,\vec{r}_4)/2] \tag{4-1}$$

where P_{ij} is the permutation operator between parallel electrons i and j. The term $\psi_A(\vec{r}_1,\vec{r}_3)$ is Schwartz's[25] 189-term Hylleraas-type atomic wave function (Table 4) for electrons 1 and 3 on nucleus A. The function U accounts for interactions between the two atoms by including terms similar to the dipole-dipole term used by Slater.[26,27]

Atomic Wave Function

Hylleraas,[22] in 1929, obtained an upper bound of -5.80648 Ry for the helium atom by using six terms in the form:

$$\psi_A(\vec{r}_1,\vec{r}_2) = \sum_{\ell,m,n} c_{\ell,m,n}\, s^{\ell}u^{m}t^{n}e^{-z's} \tag{4-2}$$

where

$$s = (r_{1A}+r_{2A})/a_B,$$
$$t = (r_{1A}-r_{2A})/a_B,$$
$$u = r_{12}/a_B . \qquad (4\text{-}3)$$

The term $r_{iA} = |\vec{r}_i - \vec{R}_A|$ is the distance between electron i and nucleus A; z' is a constant (variational parameter); and, to make $\psi_A(\vec{r}_1;\vec{r}_3)$ symmetric with respect to interchange of the electrons, t is found in even powers only. A matrix inversion is then performed to find the set of coefficients $c_{\ell,m,n}$ that give the lowest possible upper bound with the terms used. Later authors,[23-25] with the help of fast computers, tried many variations of this idea. Higher accuracy came with the inclusion of a greater number of terms in the expansion. Pekeris,[24] using 1078 terms in an expansion very similar to the above, and a 33-term recursion relation, found an essentially exact answer (-5.807448750 Ry). After this, the game became one of getting the same answer with fewer coefficients.

We used Schwartz's[25] 189-term, z' = 1.75 wave function (Table 4) which provides an upper bound to the energy of -5.80744875232 Ry. This was estimated to be above the true energy by 2.0×10^{-9} Ry.

Although Schwartz found extreme accuracy in the expectation value for the energy, we found fluctuations in the local energy to be much higher--by about a factor of 0.5×10^{6}. Since σ is proportional to fluctuations in the

Table 4. Parameters $c_{\ell,m,n}$ reading from left to right in Schwartz's 189-term atomic wave function. The "integers" (ℓ,m,n) are in the order: (0,0,0); (1,0,0), (0,1,0); (3/2,0,0), (1/2,1,0); (2,0,0), (1,1,0), (0,2,0), (0,0,2); (5/2,0,0), (3/2,1,0), (1/2,2,0), (1/2,0,2); etc. Note s includes half-integer powers while t includes even powers only.

1.000000000	-5.6762064496-1	6.3888720596-1	3.601649402	-2.018472944
-1.902943036E+1	1.2448994866+1	-1.026802958	7.6326633246-1	6.4229609446+1
-4.603636465E+1	3.361217000	-3.069398340	-1.4695365306+2	1.1210259186+2
-1.037848041E+1	1.3061001836+1	3.436179707	-4.799940197	2.3997598836+2
-1.923960173E+2	2.7701351156+1	-3.9048357786+1	-1.6950780376+1	2.3748375566+1
-2.885316167E+2	2.4143760476+2	-5.5318990456+1	8.1107442316+1	4.5253724766+1
-6.6984190336+1	-2.913505181	6.129375841	1.7249803986-1	2.5999994126+2
-2.2606942106+2	7.9998113686+1	-1.2053856296+2	-7.9829786826+1	1.2676170206+2
1.2159695916+1	-2.5797646116+1	-8.2556647386-1	-1.7706339236+2	1.5943721666+2
-8.3941326106+1	1.3059784476+2	9.8486700956+1	-1.6773360236+2	-2.4393856756+1
5.3035970906+1	1.445389874	7.0064846106-1	-1.951982372	2.7995972766-1
9.1164791916+1	-8.4739310806+1	6.4200115836+1	-1.0388516226+2	-8.6830790746+1
1.5780209866+2	3.0049289926+1	-6.7381262366+1	-1.252469963	-2.213914998
6.162512424	-8.2108782216-1	-3.5214378086+1	3.3664457346+1	-3.5729764546+1
6.0477680026+1	5.4985966576+1	-1.0593888486+2	-2.4526976046+1	5.6799620586+1
4.7039980256-1	3.120412013	-8.754168625	1.199648526	-4.6210985366-2
1.7249755926-1	-8.8799092376-2	-7.2123914986-4	1.0026709366+1	-9.812668097
1.4299215536+1	-2.5402662246+1	-2.4792123406+1	5.0295540866+1	1.3568851886+1
-1.2384200306+1	8.7169916756-2	-2.5119109999	7.138739963	-1.066658882
9.9431791676-2	-3.6929377066-1	1.8583508816-1	1.9216030826-3	-2.038107176
2.029710542	-4.005407440	7.476147452	7.758764735	-1.6463484396+1
-5.036841074	1.2350552826+1	-1.7216073466-1	1.234795512	-3.562579879
5.9486141166-1	-8.7370719416-2	3.2453519076-1	-1.6472173456-1	-1.1183573096-3
7.3150652656-4	-3.8167808196-3	4.3647459626-3	-8.2152360686-4	2.7932444706-1
-2.8089976436-1	7.4484606876-1	-1.459459700	-1.599800986	3.528033111
1.201623207	-3.016330989	7.6356020816-2	-3.6778961226-1	1.077487048
-2.0220133286-1	3.8957991436-2	-1.4526451596-1	7.5941626056-2	-1.8243949606-5
-8.9711807906-4	4.6753542676-3	-5.2373884806-3	9.6689827066-4	-2.3087778696-2
2.3215602306-2	-8.2549489396-2	1.6939848776-1	1.9519281606-1	-4.4467430966-1
-1.6640636696-1	4.2604281166-1	-1.5664652286-2	6.1099137956-2	-1.8157351906-1
3.8172563276-2	-8.7680715536-3	3.2885680406-2	-1.7931085916-2	1.6656153716-4
3.6951035226-4	-1.9161235746-3	2.1546164966-3	-4.1033436556-4	-1.6514097516-6
1.3031091476-5	-3.1681708706-5	1.7891931396-5	-3.1321090606-7	8.6784687846-4
-8.6160832106-4	4.1240288156-3	-8.8321317646-3	-1.0661845066-2	2.4948949076-2
1.0159801236-2	-2.6432772596-2	1.2764041326-3	-4.3479947126-3	1.3084126796-2
-3.0604596986-3	7.9331395996-4	-2.9943432696-3	1.7143457306-3	-3.3471263146-5
-5.1012646716-5	2.6349490656-4	-3.0167794486-4	6.1199365326-5	6.4742816046-7
-5.0474039766-6	1.2081289816-5	-6.7514854256-6	1.1616915516-7	

local energy, and since the contribution to these fluctuations from the atomic wave function is sizable (37% of σ^2 at the well minimum), it was profitable to use the most accurate atomic wave function available (Schwartz's 189-term ψ). These 189 coefficients were held fixed during the minimization of σ^2. Because they were not varied, and because ψ_A (and its first and second derivatives) were easy to store along with $\tilde{x}$ and $w(\tilde{x})$, it was necessary to evaluate these only once over $\{\tilde{x}\}_{min}$.

Interaction Terms

The function U, which accounts for interactions between the two atoms, may be broken into a sum of pair interactions with electrons from opposite nuclei:

$$U(\vec{r}_1,\vec{r}_3;\vec{r}_2,\vec{r}_4) = u(\vec{r}_1;\vec{r}_2) + u(\vec{r}_1;\vec{r}_4) + u(\vec{r}_3;\vec{r}_2) + u(\vec{r}_3;\vec{r}_4) . \quad (4\text{-}4)$$

The functions $u(\vec{r}_i;\vec{r}_j)$ are given as

$$u(\vec{r}_i;\vec{r}_j) = \sum_{\nu=0}^{3} v_\nu(\vec{r}_i;\vec{r}_j) + e(\vec{r}_i;\vec{r}_j) \quad (4\text{-}5)$$

where the v_ν terms (defined next paragraph) are different orders in the expansion of the interaction potential energy between the two atoms. This interaction potential energy can also be put into the form of Eq. (4-4):

$$V(\vec{r}_1,\vec{r}_3;\vec{r}_2,\vec{r}_4) = v(\vec{r}_1;\vec{r}_2) + v(\vec{r}_1;\vec{r}_4) + v(\vec{r}_3;\vec{r}_2) + v(\vec{r}_3;\vec{r}_4) \qquad (4\text{-}6)$$

where

$$v(\vec{r}_i;\vec{r}_j) = 1/R_{AB} - 1/r_{iB} - 1/r_{jA} + 1/r_{ij} . \qquad (4\text{-}7)$$

Since wave functions generally have the highest amplitude in the regions of low potential energy, we wanted somehow to incorporate into the wave function the ability to respond flexibly to this potential. It would have been difficult, however, to put $v(\vec{r}_i;\vec{r}_j)$ directly into $u(\vec{r}_i;\vec{r}_j)$ due to the delta functions in this term:

$$\nabla^2 \frac{1}{r} = -4\pi\delta(\vec{r}) . \qquad (4\text{-}8)$$

It was easy, though, to include into $u(\vec{r}_i;\vec{r}_j)$ a function very similar to $v(\vec{r}_i;\vec{r}_j)$:

$$v_0(\vec{r}_i;\vec{r}_j) = [(R_{AB}^2+\alpha)^{-1/2} - (r_{iB}^2+\alpha)^{-1/2} - (r_{jA}^2+\alpha)^{-1/2} + (r_{ij}^2+\alpha)^{-1/2}] f_0(r_{iA}) f_0(r_{jB}) . \qquad (4\text{-}9)$$

It can be seen that for $\alpha = 0$ the expression in brackets is equal to $v(\vec{r}_i;\vec{r}_j)$. The variable parameter α, given in Table 5, was added to eliminate the singularities mentioned above and had little effect otherwise. The function $f_0(r_{iA})$

was included to allow the importance of this term to vary with the electron-nuclear distances.

We were able to include similar terms into $u(r_i;r_j)$ by expanding the distances in Eq. (4-7) in a Taylor series about R_{AB}:

$$\frac{1}{r_{iB}} = [x_{iA}^2+y_{iA}^2+(R_{AB}-z_{iA}^2)]^{-1/2} = \frac{1}{R_{AB}}\left[1-2\frac{z_{iA}}{R_{AB}}+\frac{r_{iA}^2}{R_{AB}^2}\right]^{-1/2},$$

$$\frac{1}{r_{jA}} = \frac{1}{R_{AB}}\left[1+2\frac{z_{jB}}{R_{AB}}+\frac{r_{jB}^2}{R_{AB}^2}\right]^{-1/2},$$

$$\frac{1}{r_{ij}} = \frac{1}{R_{AB}}\left[1+2\frac{(z_{jB}-z_{iA})}{R_{AB}}+\frac{(\vec{r}_{iA}+\vec{r}_{jB})^2}{R_{AB}^2}\right]^{-1/2}. \qquad (4\text{-}10)$$

The nuclei are located on the z-axis and x_{iA}, y_{iA}, z_{iA} are the x, y, z components of $\vec{r}_{iA}$, etc. Terms multiplied by equal powers of R_{AB} were then grouped together. The first surviving term (proportional to R_{AB}^{-3}) is the dipole-dipole (D-D) interaction:

$$v_1(\vec{r}_i;\vec{r}_j) = (x_{iA}x_{jB}+y_{iA}y_{jB}-2z_{iA}z_{jB})f_1(r_{iA})f_1(r_{jB}). \qquad (4\text{-}11)$$

Again, the function f_1 was added to allow for greater flexibility in the wave function. Similarly, the dipole-quadrupole (D-Q) term (proportional to R_{AB}^{-4}) is given as

$$v_2(\vec{r}_i;\vec{r}_j) = [r_{iA}^2 z_{jB} - r_{jB}^2 z_{iA} + (z_{iA} - z_{jB})$$

$$(2x_{iA}x_{jB} + 2y_{iA}y_{jB} - 3z_{iA}z_{jB})]f_2(r_{iA})f_2(r_{jB}) \cdot \tag{4-12}$$

The quadrupole-quadrupole (Q-Q) term is

$$v_3(\vec{r}_{iA};\vec{r}_{jB}) = \{6\vec{r}_{iA}\cdot\vec{r}_{jB}[\vec{r}_{iA}\cdot\vec{r}_{jB} - r_{iA}^2 - r_{jB}^2 + 5(z_{iA} - z_{jB})^2]$$

$$- 15(z_{iA}^2 r_{jB}^2 + z_{jB}^2 r_{iA}^2$$

$$+ z_{iA}z_{jB}[30(r_{iA}^2 + r_{jB}^2) - 70(z_{iA}^2 + z_{jB}^2) + 105z_{iA}z_{jB}]$$

$$+ 3r_{iA}^2 r_{jB}^2\}f_3(r_{iA})f_3(r_{jB}) \cdot \tag{4-13}$$

The functions f_ν are splines of the form

$$f(r) = \sum_{i=1}^{6} a_i(\xi_i - r)_+^3 \tag{4-14}$$

where

$$(x)_+ = \begin{cases} x, & x>0 \\ 0, & x<0 \end{cases} \cdot \tag{4-15}$$

The functions f_ν therefore become zero for large values of their arguments and thereby tend to make the v_ν terms a function of the interactions of the electrons "on" nucleus A with those "on" nucleus B. Since the v_ν functions appear exponentially in $\Psi(\tilde{x})$, this feature is necessary for the

wave function to obey the boundary condition that, for all i:

$$\lim_{r_{iA}\to\infty}[r_{iA}\Psi(\tilde{x})] \to 0 . \qquad (4\text{-}16)$$

Finally, to allow for close encounters of electrons not already accounted for by the atomic wave functions, the term

$$e(\vec{r}_i;\vec{r}_j) = e(r_{ij}) = \sum_{i=1}^{2} a_i'(\xi_i' - r_{ij})_+^3 \qquad (4\text{-}17)$$

was added to $u(\vec{r}_i;\vec{r}_j)$. Consider the region of $\tilde{x}$ where r_{ij} is much smaller than all other distances. The dominant term in the local energy, $H\Psi(\tilde{x})/\Psi(\tilde{x})$, then becomes

$$H\Psi(\tilde{x})/\Psi(\tilde{x}) \approx [(2/r_{ij} - \nabla_i^2 - \nabla_j^2)\exp(-e(r_{ij})/2)]\exp[+e(r_{ij})/2]$$

$$= 2/r_{ij} + 2e'(r_{ij})/r_{ij} + e''(r_{ij})/2 . \qquad (4\text{-}18)$$

The derivative $e'(0)$, therefore, was set equal to -1 to cancel the singularities and to make the local energy more constant. The coefficients a_i, a_i', and knots ξ_i', given in Table 5, are parameters with respect to which the trial wave functions were optimized (as described in Ch. III). The knots ξ_i were kept fixed at 2.0, 4.0, 6.0, 7.0, 8.0, and 9.0 a_B.

Table 5. Variable parameters for Eqs. (4-9), (4-14), and (4-17) which, together with Schwartz's parameters, determine the wave functions.

R	4.5	5.0	5.6	6.6	7.5	9.0	15.0
D-D	-4.70892E-2	-1.55826E-2	-4.29590E-4	7.47165E-3	-4.20371E-2	-6.14472E-2	-6.43464E-3
COEFS	4.15209E-2	2.22784E-2	-1.07552E-2	1.03047E-2	6.39287E-2	1.75601E-1	2.07885E-2
	5.44734E-2	7.87565E-2	1.66395E-1	9.09020E-2	-7.21025E-3	-3.20039E-1	1.24759E-1
	-2.77328E-1	-3.22569E-1	-4.37159E-1	-3.99165E-1	-3.01037E-1	1.74799E-2	-5.01352E-1
	3.16033E-1	3.57847E-1	4.20820E-1	4.58833E-1	4.27316E-1	3.39207E-1	5.60043E-1
	-1.11122E-1	-1.25005E-1	-1.38560E-1	-1.62970E-1	-1.61986E-1	-1.66929E-1	-1.96674E-1
D-D	-4.09052E-3	-1.26733E-3	-2.98240E-5	8.29648E-4	2.04350E-4	7.58871E-4	----------
COEFS	4.43566E-3	2.02803E-3	-7.12454E-4	7.13705E-4	1.88541E-3	-3.55547E-3	----------
	6.37596E-3	7.98084E-3	1.27601E-2	4.99981E-3	2.60785E-4	5.96739E-3	----------
	-3.07022E-2	-3.14954E-2	-3.41041E-2	-2.18727E-2	-1.05977E-2	1.26135E-3	----------
	3.44949E-2	3.45461E-2	3.30412E-2	2.51882E-2	1.44256E-2	-8.56932E-3	----------
	-1.20574E-2	-1.20107E-2	-1.09097E-2	-8.95646E-3	-5.40133E-3	3.96460E-3	----------
D-Q	-2.48064E-3	-5.58420E-4	-1.24930E-5	2.46903E-4	-6.17894E-4	2.51864E-4	----------
COEFS	2.14743E-3	9.23865E-4	-2.45704E-4	3.87116E-4	6.23788E-4	-7.19779E-4	----------
	1.79425E-3	2.49016E-3	4.70030E-3	1.17905E-3	7.82791E-5	1.31182E-3	----------
	-1.07968E-2	-1.16825E-2	-1.27553E-2	-7.67043E-3	-2.90100E-3	-7.16191E-5	----------
	1.24540E-2	1.33609E-2	1.24277E-2	9.61423E-3	3.90153E-3	-1.39043E-3	----------
	-4.38544E-3	-4.72081E-3	-4.11347E-3	-3.52716E-3	-1.45322E-3	6.84249E-4	----------
D-D	-5.61605E-4	-7.67956E-5	-1.54890E-6	-1.47917E-5	-4.36035E-4	----------	----------
COEFS	2.74914E-4	7.65038E-5	-5.93223E-6	-4.83552E-5	1.07403E-3	----------	----------
	2.26788E-4	2.78610E-4	4.34381E-4	-6.50308E-5	-3.38105E-4	----------	----------
	-1.31339E-3	-1.27416E-3	-1.27742E-3	6.84898E-4	-3.77891E-3	----------	----------
	1.50163E-3	1.44960E-3	1.27955E-3	-9.08712E-4	5.55812E-3	----------	----------
	-5.26785E-4	-5.11162E-4	-4.28494E-4	3.39563E-4	-2.12206E-3	----------	----------
E-E	-2.81922	1.32947E 1	4.98646E-1	-3.01473	-3.27098	-1.94296	-1.96270
COEFS	3.20660	-8.09490E-1	-1.09985	2.39107	1.86162	1.13710	1.13710
E-E	1.13759	2.60407E-1	1.25888	9.34652E-1	5.01013E-1	8.26368E-1	8.22202E-1
KNOTS	1.11433	8.37815E-1	6.44535E-1	1.11414	7.87467E-1	1.20830	1.20830

Properties of the Interaction Terms

Consider the functions f_ν normalized such that

$$u(\vec{r}_i;\vec{r}_j) = v_0(\vec{r}_i;\vec{r}_j) + R_{AB}^{-3}(\text{D-D terms})f_1(r_{iA})f_1(r_{jB}) + R_{AB}^{-4}(\text{D-Q terms})f_2(r_{iA})f_2(r_{jB}) + R_{AB}^{-5}(\text{Q-Q terms})f_3(r_{iA})f_3(r_{jB}) . \quad (4\text{-}19)$$

Here the D-D, D-Q, and Q-Q terms are obtained directly from the expansion of the interaction potential without being multiplied by any constants. By comparing this to the similar expansion of the interaction potential,

$$v(\vec{r}_i;\vec{r}_j) = R_{AB}^{-3}(\text{D-D terms}) + R_{AB}^{-4}(\text{D-Q terms}) + \ldots ; \quad (4\text{-}20)$$

a first guess for the functions f_1, f_2, f_3 would have them all roughly equal and roughly constant for small r. Figure 2 shows several departures from this simple behavior;

1. The contribution of the D-D term (obtained by adding the f_0 curve to the f_1 curve), D-Q term, etc. to the ratio $u(\vec{r}_i;\vec{r}_j)/v(\vec{r}_i;\vec{r}_j)$ decreases with increasing order. From the sole consideration that $u(\vec{r}_i;\vec{r}_j)$ lowers the energy by increasing $\Psi(\vec{x})$ in the regions where the potential is low, these ratios would be equal. The functions

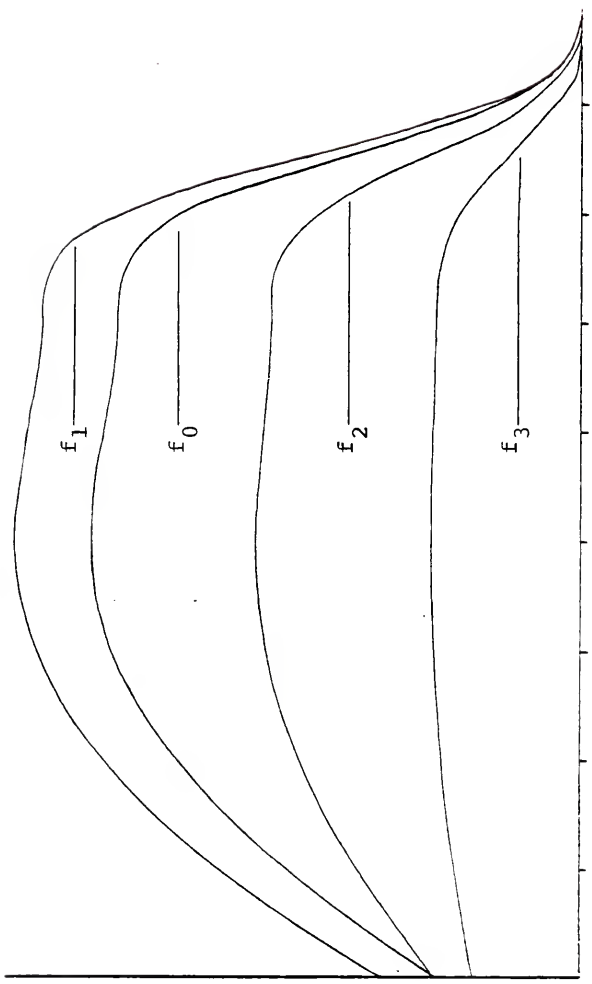

Figure 2. Relative values for the normalized functions $f_\nu(r_{iA})f_\nu(r_{jB})$ defined in Eq. (4-20) as a function of r_{iA} for the nuclear separation distance of 5.6 a_B. The radial distance r_{jB} is set here at the typical value of 1.0 a_B. The total contribution of the dipole-dipole interaction can be obtained by adding the f_1 curve to the f_0 curve; similarly, the contribution of the dipole-quadrupole effects are found by adding the f_2 curve to the f_0 curve, etc.

$u(\vec{r}_i;\vec{r}_j)$, however, also have the effect of increasing the kinetic energy. Since the kinetic energy is proportional to the second derivative of the wave function, smoother terms introduced into $u(\vec{r}_i;\vec{r}_j)$ will be more efficient at lowering the total energy. This helps to explain why the more oscillatory higher order terms in the expansion of $v(\vec{r}_i;\vec{r}_j)$ have less importance than first expected.

2. The functions f_ν decrease significantly from $r = 3\ a_B$ to $r = 0\ a_B$. Again, from the lone assumption that $u(\vec{r}_i;\vec{r}_j)$ is proportional to $v(\vec{r}_i;\vec{r}_j)$, these curves would be roughly flat. [Of course they must become zero for large r to satisfy the boundary condition at $r = \infty$, Eq. (4-16).] This decrease in the electron's response to molecular effects from the opposite atom is apparently due to the increased influence of the atomic effects from the nearer nucleus. Increased shielding as $r \to 0$ from the other electron "on" this nucleus may also be an influence in the decreased importance of these terms.
3. The functions f_ν have different shapes--becoming flatter with increasing order. It is estimated that σ was lowered by an extra 4-12% by allowing these shapes to vary independently. This may be related to the trend that, as the order of the

term increases, so does the value of r for which that term has maximum effect.

Beyond the radial distance of $r = 4\ a_B$, where the wave function is small, the exact shape of these functions has very little importance.

The molecular terms described above and included in the factor $e^{-U/2}$ [Eq. (4-1)] account for the vast bulk of the molecular behavior of the two-helium-atom system. Since the Monte Carlo standard deviation, σ, is a measure of the accuracy of the wave function, the degree to which this is true can be measured by comparing this quantity over a fixed set of MC points. At the separation distance of $R_{AB} = 5.6\ a_B$, without the inclusion of any terms in U, the standard deviation over 6400 MC points is $(R_{AB}=5.6a_B, N=6400) = 40.2 \times 10^{-5}$ Ry. With the inclusion of the molecular terms, this dispersion drops by over a factor of 10 to 3.83×10^{-5} Ry --much closer to the standard deviation of the widely separated atoms [$\sigma(R_{AB} \geq 15.0a_B, N=6400) = 2.35 \times 10^{-5}$ Ry]. Since the atomic coefficients [$c_{\ell,m,n}$ in Eq. (4-2)] are held fixed, and since the atomic and molecular contributions to variations in the local energy can be considered to be roughly independent, the total standard deviation can be written:

$$\sigma^2(R_{AB},N) \approx \sigma_A^2(N) + \sigma_M^2(R_{AB},N) \qquad (4\text{-}21)$$

where σ_A^2 and σ_M^2 are the atomic and molecular contributions to σ^2 and

$$\sigma_{Atomic}(N=6400) = \sigma(R_{AB} \geq 15.a_B, N=6400) = 2.35 \times 10^{-5} \text{ Ry} \tag{4-22}$$

for all internuclear separation distances. From this relation, we can see that σ can only increase as R_{AB} decreases due to the increasing molecular effects. This is basically caused by the form used for the wave function--that of two atoms weakly interacting through multipole forces. There will, in fact, be some point, as R_{AB} gets smaller, where the wave function used fails to accurately describe the actual eigenfunction. This is due partly to the breakdown in the accuracy of the basic form for $\Psi(\tilde{x})$ and partly due to the increasing importance of higher order terms (such as octupole-octupole) not included in U. As shown in Fig. 3, this begins at about $R_{AB} = 5.0\ a_B$ (where the potential "barrier" starts up).

For a long time we used Schwartz's 164-term atomic wave function. But, with the inclusion of v_0 and v_3 into the wave function, σ_M^2 became small enough, compared to σ_A^2, that it became advantageous to switch to his most accurate, 189-term, atomic wave function. Merely switching the atomic wave function did improve the total wave function, but a further improvement came by reoptimizing the variable (molecular) parameters with the new atomic wave function

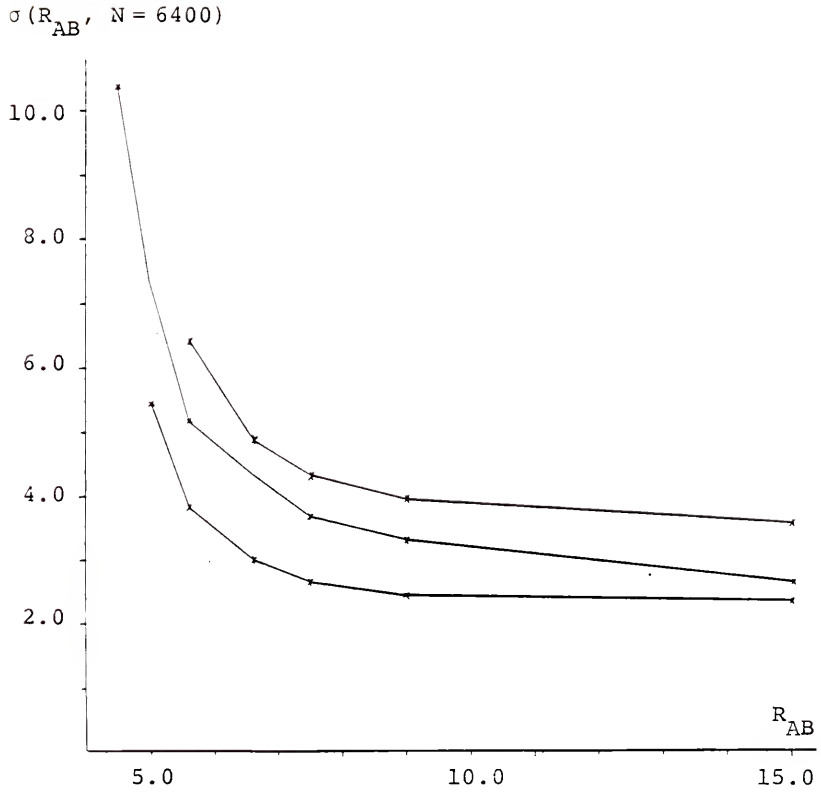

Figure 3. Standard deviations in units of 10^{-5} R_y for some trial wave functions evaluated over 6400 Monte Carlo points. Top curve is for wave functions optimized with Schwartz's 164-term atomic wave function. The middle curve is for the same wave functions but with Schwartz's 189-term function substituted for the 164-term wave function. The bottom curve is for the wave functions optimized with the 189-term atomic wave function.

(Fig. 3). It is apparent that reducing the "noise" (σ_A) from the atomic effects made it easier for the optimization routine to reduce σ_M.

Discarded Forms for the Trial Wave Function

The first wave function we constructed has the form,[16]

$$\Psi(\tilde{x}) = \exp[-U(\vec{r}_1,\vec{r}_2,\vec{r}_3,\vec{r}_4)/2]$$

$$(1-P_{12}-P_{34}+P_{12}P_{34})\psi_A(\vec{r}_1,\vec{r}_3)\psi_B(\vec{r}_2,\vec{r}_4) \qquad (4\text{-}23)$$

where the function U is completely symmetric:

$$U = \sum_{i\neq j} u(\vec{r}_i;\vec{r}_j) \; . \qquad (4\text{-}24)$$

We can see from Eq. (4-23) that the terms with the atomic wave function $\psi_A(\vec{r}_1;\vec{r}_3)$, for example, are multiplied by the factor $\exp[-u(\vec{r}_1;\vec{r}_3)/2 - u(\vec{r}_3;\vec{r}_1)/2]$. But, since electrons 1 and 3 are "on" the same atom in these terms, the factor above accounts for atomic rather than molecular effects. This forced us to make the functions f [Eq. (4-14)] to be short-ranged to prevent interference with the already accurate atomic wave functions; and, therefore, severely limited the flexibility of the form shown in Eq. (4-23).

Before the v_0 and v_3 terms were included in $\Psi(\tilde{x})$, several extended versions of the functional form were tested:

1. The functions $f_1(r)$ and $f_2(r)$ in the v_1 and v_2 terms [Eq. (4-11) and Eq. (4-12)] were generalized to $f([x^2+y^2+\varepsilon(z-\alpha)^2]^{1/2})$ where ε and α were allowed to vary with all the other variational parameters.
2. The v_1 and v_2 functions in the factor $e^{-U/2} = \exp\{-[v_1-v_2-f(r_{12})]/2\}$ were expanded to $(1-c_1v_1/2-c_2v_2/2$ + quadratic and cubic terms$)\exp[-f(r_{12})/2]$ and the coefficients c allowed to vary. Incidentally, with all the coefficients set equal to 1, the two forms above were equivalent in the energy to 9 decimal places. This is a good measure of the weakness of these Van der Waals forces on the atoms.
3. Instead of using one function for all electron-electron interactions, as in Eq. (4-17), two e-e functions were included--one for spin-parallel interactions, the other for spin-antiparallel interactions--and allowed to vary independently.
4. The factor of 2 multiplying $z_{iA}z_{jB}$ in Eq. (4-11) was allowed to vary.
5. A spline function $s(z_i) + s(z_j)$ was added to $u(\vec{r}_i;\vec{r}_j)$ [Eq. (4-5)] to allow the electrons to increase (or decrease) the probability of being in the region between the nuclei.

None of these attempted improvements gave $\Psi(\tilde{x})$ the flexibility needed at the time to lower σ which, of course, was

finally provided by the inclusion of the v_0 and v_3 functions. With much of the contribution to σ eliminated by the v_0 and v_3 terms, it might be argued that the improvements above could more effectively reduce σ further. But, in each case, these generalizations to $\Psi(\tilde{x})$ failed to lower σ at all. So, even with the inclusion of the v_0 and v_3 terms, it would be surprising if they had any effect now.

The most likely way that $\Psi(\tilde{x})$ might be improved would be to carry the expansion of the interaction potential energy to higher order. I believe that with the present wave function, however, this would be more trouble than it is worth. With the inclusion of the v_0, v_1, v_2, v_3 functions into u, even quadrupole-octupole, v_4, effects are accounted for. Including the next function, v_4, into $u(\vec{r}_i;\vec{r}_j)$ would free v_0 to adjust for v_5 effects; but, due to the very large number of terms in v_4, this would be very difficult to code and time consuming to evaluate on a computer. An estimate of the improvement made by the evaluation of further terms can be made by extrapolation. By including into the wave function the first two terms in the expansion of $v(\vec{r}_i;\vec{r}_j)$ [v_1 and v_2 from Eqs. (4-11) and (4-12)], $\sigma(R_{AB}=5.6a_B)$ was lowered by over a factor of 5; adding the next two functions [v_0 and v_3 from Eqs. (4-9) and (4-13)] into $\Psi(\tilde{x})$ lowered this by not quite another factor of 2.

CHAPTER V
DISCUSSION AND RESULTS

Differencing

For the internuclear separation distances $R_{AB} > 5.6\ a_B$, Fig. 3 shows that the standard deviations are not much greater than that due to the dispersion in the atomic wave functions. This implies that taking the difference in the energies at two different nuclear separation distances ($R_{AB} = R,R'$) will subtract out a good portion of the atomic effects; and, as long as both energies are evaluated over a very similar set of MC points, this will yield standard deviations smaller than for either individual energy. The key to actually doing this is the simple relationship

$$\int_{-\infty}^{\infty} f[g(x)]g'(x)dx = \int_{-\infty}^{\infty} f(y)dy \qquad (5\text{-}1)$$

which means that the free function g can be used to make

$$\Delta E = \frac{\int \Psi(R,\tilde{x}) H \Psi(R,\tilde{x}) d\tilde{x}}{\int \Psi^2(R,\tilde{x}) d\tilde{x}} - \frac{\int \Psi[R',g(\tilde{x})] H \Psi[R',g(\tilde{x})] d\tilde{x}}{\int \Psi^2[R',g(\tilde{x})] d\tilde{x}} \qquad (5\text{-}2)$$

as smooth as possible. The function, g, is a product of one-dimensional functions that serve to make all the spreading and contracting occur in the relatively unimportant region between the nuclei:

$$g(\tilde{x}) = \hat{g}(\vec{r}_1)\hat{g}(\vec{r}_2)\hat{g}(\vec{r}_3)\hat{g}(\vec{r}_4) \qquad (5\text{-}3)$$

where

$$\hat{g}(\vec{r}_i) = \hat{g}(z_i) = \begin{cases} (2+R'-R)z_i/2, & |z_i| \leq 1 \\ z_i + (R'-R)/2, & z_i \geq 1 \\ z_i - (R'-R)/2, & z_i \leq -1 \end{cases} \qquad (5\text{-}4)$$

and z_i is the distance of the i'th electron from the plane bisecting the nuclei. For $|z_i| \geq 1$, this transformation makes the distances of $\hat{g}(\vec{r}_i)$ and $\vec{r}_i$ to the nearest nucleus exactly equal. Note that $\hat{g}(z)-z$ looks like a ramp. Results from the final MC sums that determined the energies, these energy differences, and the corresponding standard deviations are reported in Table 6. One sum over 782,000 MC points was taken with R = 5.6 a_B and transformed to the distances of R' = 6.6, 7.5, 9.0, and 15.0 a_B; another was taken over 377,000 points with R = 4.5 a_B and transformed to R' = 5.0 and 5.6 a_B.

This differencing technique was also used during the optimization of the wave functions. Since the form of the wave function is that of two atoms weakly interacting through molecular terms which contain all the variable parameters, optimizing these parameters can yield a standard deviation no less than that for the two widely separated atoms:

Table 6. Energies, energy differences, and kinetic energies with standard deviations in units of 10^{-5} Ry.

R_i	$E(R_i)$	$E(R_i)-E(R_{i+1})$	$E(R_i)-E(R_{i+2})$	$K(R_i)-K(15)$
4.5	41.35 ± 1.29	41.19 ± 0.68	48.52 ± 0.93	643. ± 25.
5.0	0.16 0.82	7.33 0.34	---- ----	216. 19.
5.6	-7.16 0.33	-2.92 0.22	-5.01 0.24	61. 16.
6.6	-4.24 0.29	-2.09 0.13	-3.43 0.16	19. 14.
7.5	-2.15 0.25	-1.34 0.07	-1.99 0.13	17. 12.
9.0	-0.81 0.22	-0.65 0.07	---- ----	15. 9.
15.0	-0.16 0.21	---- ----	---- ----	---- ---

$$\sigma^2 \approx \sigma_A^2+\sigma_M^2 \geq \sigma_A^2 \quad . \tag{5-5}$$

Since optimizing the set of parameters can do nothing to lower σ_A^2, this contribution to σ^2 is an undesirable "noise" that the minimization routine must sieve through. Most of σ_A^2 can be eliminated, however, by minimizing the standard deviation, σ_d, of (rather than the energy) the difference of two energies. It can be seen from Fig. 3 that there is an increasing (as R_{AB} decreases) contribution to σ^2 that the variable parameters cannot get rid of. Consequently, this, too, is extraneous noise and should be subtracted out along with σ_{Atomic}. The wave functions at each internuclear separation were, therefore, optimized by minimizing σ_d with respect to the next wave function. For example, $\Psi(R_{AB}=4.5a_B,\tilde{x})$ was found by minimizing $\sigma_d(R=4.5a_B,R'=5.0a_B)$, $\Psi(R_{AB}=6.6a_B,\tilde{x})$, by minimizing $\sigma_d(R=6.6a_B,R'=7.5a_B)$, etc. When tested over a set of MC points independent from $\{\tilde{x}\}_{min}$, wave functions found by minimizing σ_d were found to have significantly lower σ_d's and lower σ's than wave functions found by minimizing σ.

Curve Fit

Values for the energies from Eq. (2-20), the energy differences mentioned above, and the corresponding standard deviations (Table 6) were used in a curve fit for minimizing

$$\chi^2 = \sum_{i=1}^{7} \left(\frac{E_{Fit}(R_i) - E_{MC}(R_i)}{\lambda_i \sigma_{MC}(R_i)} \right)^2$$

$$+ \sum_{i=1}^{15} \left(\frac{\Delta E_{Fit}(R_i, (\Delta R)_i) - \Delta E_{MC}(R_i, (\Delta R)_i)}{\hat{\lambda}\hat{\sigma}_{MC}(R_i, (\Delta R)_i)} \right)^2 \qquad (5\text{-}6)$$

where

$$\Delta E(R_i, (\Delta R)_i) = E(R_i + (\Delta R)_i) - E(R_i) , \qquad (5\text{-}7)$$

and $\hat{\sigma}_{MC}(R_i, (\Delta R)_i)$ is the standard deviation of $\Delta E_{MC}(R_i, (\Delta R)_i)$. The factors λ_i and $\hat{\lambda}_i$ made it possible to test for the relative importance of these terms. The best final result had error bars about 3% less than would have been found with $\lambda_i = \hat{\lambda}_i = 1$. This yields

$$E_{Fit}(R) = E_{BFL}(R) = \sum_{i=1}^{5} c_i (R-4.0)_+^2 (k_i - R)_+^2 \qquad (5\text{-}8)$$

with $\{c_i\} = +0.558029 \times 10^{-5}$, -0.592108×10^{-6}, -0.140675×10^{-6}, -0.349129×10^{-7}, -0.709027×10^{-9} Ry/a_B^4 and $\{k_i\} = 5.6$, 6.6, 7.5, 9.0, 15.0 a_B. The term $E_{BFL}(R)$ is the experimental curves of Burgmans, Farrar, and Lee[4] which was used because it goes to the accepted[30] limits at small ($R < 4.0\ a_B$) and large ($R > 10.0\ a_B$) internuclear distances. The energies $E_{Fit}(R)$ and $E_{BFL}(R)$ for $5.0\ a_B < R < 9.5\ a_B$ are given in Table 7 and Fig. 4.

Table 7. Energies with standard deviations from the curve fit along with the experimental results of Burgmans, Farrar, and Lee in units of 10^{-5} Ry.

R	E_{BFL}	E_{MC}
5.0	0.36	0.26 ± 0.52
5.1	-2.35	-2.51 0.48
5.2	-4.22	-4.44 0.43
5.3	-5.44	-5.73 0.38
5.4	-6.18	-6.53 0.34
5.5	-6.57	-6.96 0.31
5.6	-6.70	-7.10 0.30
5.7	-6.65	-7.04 0.29
5.8	-6.48	-6.87 0.28
5.9	-6.23	-6.61 0.27
6.0	-5.93	-6.30 0.26
6.1	-5.60	-5.94 0.25
6.2	-5.25	-5.57 0.24
6.3	-4.90	-5.20 0.23
6.4	-4.54	-4.82 0.22
6.5	-4.20	-4.46 0.21
6.6	-3.88	-4.13 0.19
6.7	-3.59	-3.82 0.18
6.8	-3.31	-3.53 0.17
6.9	-3.06	-3.27 0.16
7.0	-2.82	-3.02 0.15
7.1	-2.60	-2.79 0.14
7.2	-2.40	-2.57 0.14
7.3	-2.22	-2.38 0.13
7.4	-2.05	-2.20 0.13
7.5	-1.89	-2.04 0.12
7.6	-1.75	-1.89 0.12
7.7	-1.62	-1.75 0.12
7.8	-1.50	-1.63 0.11
7.9	-1.39	-1.51 0.10
8.0	-1.29	-1.41 0.10
8.1	-1.20	-1.31 0.09
8.2	-1.12	-1.22 0.09
8.3	-1.04	-1.13 0.08
8.4	-0.97	-1.05 0.08
8.5	-0.90	-0.98 0.08
8.6	-0.83	-0.91 0.07
8.7	-0.78	-0.84 0.07
8.8	-0.72	-0.79 0.07
8.9	-0.67	-0.74 0.07
9.0	-0.63	-0.69 0.07
9.1	-0.58	-0.65 0.07
9.2	-0.55	-0.61 0.07
9.3	-0.51	-0.57 0.07
9.4	-0.48	-0.54 0.07
9.5	-0.45	-0.51 0.07

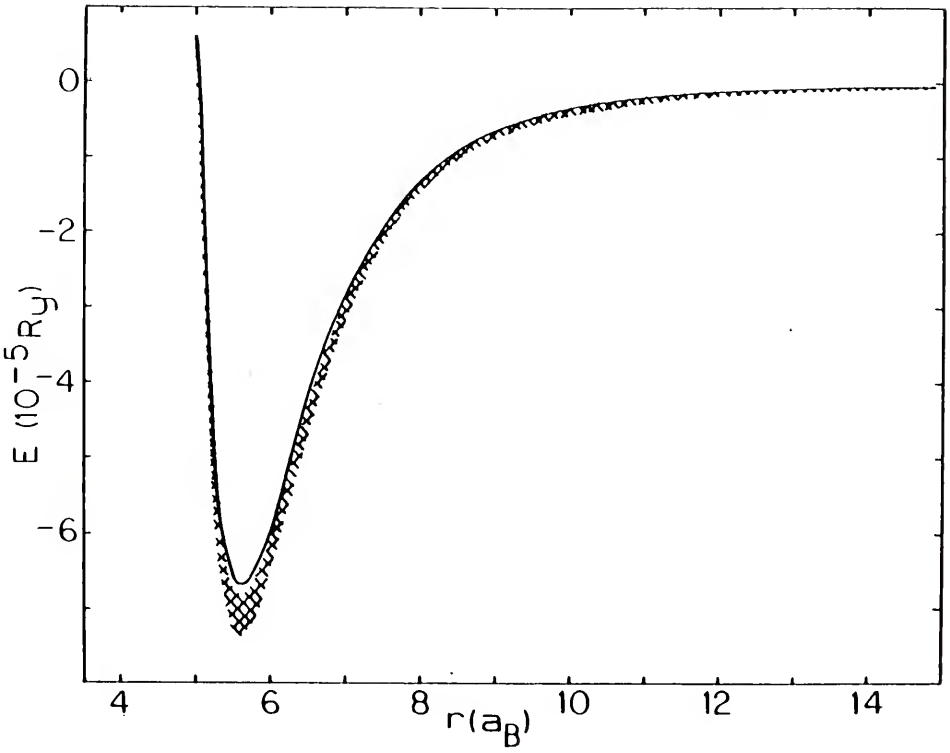

Figure 4. Upper bound to the Born-Oppenheimer potential (hatched curve). The curve is two of our standard deviations wide. Solid line is the experimental curve of Burgmans, Farrar, and Lee.

In a manner similar to the derivation of Eq. (2-18), the Monte Carlo standard deviation in the fit can be found by squaring the equation

$$E_N - E_\infty = -\sum_{i=1}^{N} \delta_i \tag{5-9}$$

and taking an ensemble average. The quantity E_∞ is the true ($N = \infty$) energy for the trial wave function and δ_i is the error introduced into E_N by omitting the i'th point. This yields the formula for the standard deviation of the energy at a nuclear separation distance R on the curve:

$$\langle (E_N - E_\infty)^2 \rangle = \left\langle \sum_{i=1}^{N} \sum_{j=1}^{N} \delta_i \delta_j \right\rangle = \left\langle \sum_{i=1}^{N} \delta_i^2 \right\rangle \approx \sum_{i=1}^{N} \delta_i^2 . \tag{5-10}$$

Our estimate for this quantity (Table 7) was found by breaking our total run into 1159 partial runs, each representing a point i, and redoing the curve fit successively leaving each of these out.

Our outermost energy $(-0.160 \pm 0.208) \times 10^{-5}$ Ry at $R = 15.0\ a_B$ was in good agreement with the accepted asymptotic results (-0.027×10^{-5} Ry, as found in BFL and references therein). Since the other theories are more accurate in this region, our energy was made equal to this value at this point. Through differencing, this had the effect of slightly raising the rest of the curve. This also reduced the standard deviation in the fit to about equal to the standard deviation in the difference between

these energies from the energy at R = 15.0 a_B. Consequently, for the internuclear separations for which E(R) was evaluated directly, the curve fit (Table 7) gives a better estimate to these energies than the direct calculations (Table 6) since more information went into it.

Between these seven calculated energies there was error introduced by the looping of the fitting function. It was found that variation of the form of the fitting function and of the knot locations k_i produced variations in E(R) between these points on the order of 0.3 standard deviations. The directly calculated points, however, were independent of these variations (to within 0.1σ). The present fit was used because it looked smooth and it had the qualitative features expected from the input.

Conclusion

The standard deviations of the curve in Fig. 4 and Table 7 are all smaller than the total electronic energy by factors less than 5.0×10^{-7}. We have therefore demonstrated the ability to produce extremely accurate energies using Monte Carlo techniques. Although the accuracy achieved here was largely due to Schwartz's extremely precise atomic wave function, chemical accuracy for most systems is around 0.1 Ry--more than 10^3 times the standard deviations reported herein.

As befits the term Monte Carlo, these calculations are inherently risky and expensive since the best trial wave

function is found without knowing the energy bound that it predicts. The upper bound calculated here is lower than the BFL[4] result in the region $R_{AB} \geq 5.1\ a_B$ with a difference of 0.40×10^{-5} Ry (1.33 standard deviations) at the potential minimum. Since this is at the 82% confidence level, this curve should be considered in agreement with the BFL result with just a hint that the true curve is deeper than the BFL curve. In the region $4.5\ a_B \leq R_{AB} \leq 5.0\ a_B$, however, our result is above the BFL curve with a difference of 1.07×10^{-5} Ry (0.83 standard deviations) at $R_{AB} = 4.5\ a_B$. In fact, the calculated energy difference from Table 6 between $E(R_{AB}=4.5a_B)$ and $E(R_{AB}=5.6a_B)$ is $48.52 \pm 0.93 \times 10^{-5}$ Ry or 1.66 standard deviations above the corresponding BFL value. This difference is probably due to the relative inaccuracy of our wave functions in the region of small R_{AB}. And yet, Fig. 4 shows that this difference is unimportant in determining the curve: it is so steep in this area that a difference in the energy produces an extremely small change in the position of the potential barrier. It is, therefore, very hard for the scattering experiments (since they are sensitive to the overall shape of the potential) to measure the energy in this region to this accuracy.

CHAPTER VI
CHECKS AND CORRECTIONS

Is $\tilde{x}$-space Sampled Adequately?

As long as the standard deviation, σ, is accurate, there is a definite bound on how far off the energy can be. This standard deviation may, however, be artificially small if a region of the position space $\tilde{x}$ is inadequately sampled. One way to be sure that this is not the case is to test the convergence property [Eq. (1-3)] that σ should have for a large number, N, of MC points. If new points are picked in previously inadequately sampled regions, then the standard deviation for a longer run will be larger than that predicted from Eq. (1-3). This is due to a small weight function w_i for these points appearing in the sums of Eq. (2-22). To check this, the standard deviations of various energies for ten partial runs with N = 2000 and N = 10,000 were compared to the average of ten partial runs with N = 50,000. From the runs with N = 2000 and N = 10,000, the values predicted for σ(N=50000) were equal to σ(N=50000) to within uncertainties of 2% and 1%.

Integration by Parts

An integration by parts of the integral in Eq. (2-19) yields an integral which becomes

$$E = \frac{\sum_{i=1}^{N} [\square\Psi(\tilde{x}_i)\cdot\square\Psi(\tilde{x}_i) + V\Psi^2(\tilde{x}_i)]/w_i}{\sum_{i=1}^{N} \Psi^2(\tilde{x}_i)/w_i} \qquad (6\text{-}1)$$

where $\square$ is the electronic gradient operator. As a check for coding errors this expression was evaluated concurrently with Eq. (2-20) over the same set of $\tilde{x}_i$. To see the difference in Eq. (6-1) from Eq. (2-20), consider the kinetic energy term for the simple atomic helium wave function

$$\Psi = e^{-2(r_1+r_2)} . \qquad (6\text{-}2)$$

For Eq. (6-1) this is

$$\square\Psi\cdot\square\Psi = 8e^{-4(r_1+r_2)} , \qquad (6\text{-}3)$$

with nothing to cancel the integrable singularities in the potential. The equivalent term for Eq. (2-20) is

$$-\Psi\square^2\Psi = 4(1/r_1 + 1/r_2 - 2)e^{-4(r_1+r_2)} , \qquad (6\text{-}4)$$

with the 4/r terms canceling with potential terms making the local energy relatively constant. This, of course, makes Eq. (2-20) many times more precise using our methods than Eq. (6-1). Since these two equations are different and also weight differently the different regions of space, agreement between them is a good test for errors and for adequate sampling of the space of $\tilde{x}$. These values from Eq. (6-1) are in agreement with the results from Eq. (2-20)

(Table 6) but with standard deviations on the order of 0.019 Ry. Energy differences can also be calculated using Eq. (6-1). This lowers the standard deviation to about 0.003 Ry--still in agreement with the very accurate results from Eq. (2-20).

Comparison to the Hydrogen Molecule

As a check on the weight functions and differencing technique, our first 80,000 MC points at R = 6.6 a_B were scaled to the system of two infinitely separated H_2 molecules. The relevant expression is

$$E_{H-H} =$$

$$\frac{1}{2}\,\frac{\int \phi_{AB}(\vec{r}_1,\vec{r}_2)\,\phi_{CD}(\vec{r}_3,\vec{r}_4)\,H\,\phi_{AB}(\vec{r}_1,\vec{r}_2)\,\phi_{CD}(\vec{r}_3,\vec{r}_4)\,d\vec{r}_1 d\vec{r}_2 d\vec{r}_3 d\vec{r}_4}{\int \phi_{AB}^2(\vec{r}_1,\vec{r}_2)\,\phi_{CD}^2(\vec{r}_3,\vec{r}_4)\,d\vec{r}_1 d\vec{r}_2 d\vec{r}_3 d\vec{r}_4} \tag{6-5}$$

where ϕ_{AB} is James and Coolidge's[31] 11-term trial H_2 wave function for electrons 1 and 2 on nuclei A and B, and H is the sum of the Hamiltonians for the two independent H_2 molecules:

$$H = H_{AB12} = H_{CD34}\,, \tag{6-6}$$

where

$$H_{AB12} = \nabla_1^2 + \nabla_2^2 - \frac{2}{r_{1A}} - \frac{2}{r_{1B}} - \frac{2}{r_{2A}} - \frac{2}{r_{2B}} + \frac{2}{r_{12}} + \frac{2}{R_{AB}}\,. \tag{6-7}$$

Note the absence of any cross-potential terms. The Monte Carlo estimate for this energy was

$$E = \frac{\sum_{i=1}^{80000} \frac{1}{w_i}(1+P_{23}+P_{24})\phi_{AB}(\vec{r}_{1i},\vec{r}_{2i})\phi_{CD}(\vec{r}_{3i},\vec{r}_{4i})(H_{AB12} + H_{CD34})\phi_{AB}(\vec{r}_{1i},\vec{r}_{2i})\phi_{CD}(\vec{r}_{3i},\vec{r}_{4i})}{2\sum_{i=1}^{80000} \frac{1}{w_i}(1+P_{23}+P_{24})\phi_{AB}^2(\vec{r}_{1i},\vec{r}_{2i})\phi_{CD}^2(\vec{r}_{3i},\vec{r}_{4i})} \qquad (6\text{-}8)$$

where the $\vec{r}_i$ are electron positions originally picked with respect to the 2He system, then scaled to the $2H_2$ system using the differencing technique in Ch. V; and w_i is the weight function used to pick these original points multiplied by a scaling factor. The permutations were used to include all possible combinations with the same weight function and had the effect of reducing the standard deviation by smoothing the integral and by increasing the number of evaluations. At the nuclear separations ($R_{AB} = R_{CD}$) of 1.2, 1.4, 1.5, and 1.7 a_B the energies were -4.416 ± 0.013, -4.674 ± 0.013, -4.612 ± 0.014, and -4.350 ± 0.010 eV compared with James and Coolidge's -4.41, -4.68, -4.63, and -4.35 eV.

Difference between <H> and the Eigenenergy

Since our trial wave functions are not exact eigenfunctions, the variational bounds would be above the correct eigenfunctions if the calculations were made over an infinite number of Monte Carlo evluations. An estimate of

the size of this effect can be made by assuming that this error is proportional to σ_M^2 [Eq. (4-21)]. During our search for the best wave function [lowest $\sigma_M(N=1000)$], energies were found for two of the earlier forms of the wave function at $R_{AB} = 5.6\ a_B$. A straight line fitted through the plot of E (-5.07 ± 0.60, -5.95 ± 0.41, $-.716 \pm 0.33\times10^{-5}$ Ry) vs. $\sigma_M^2(N=1000)$ (10.0, 1.8, 0.9×10^{-8} Ry^2) for these wave functions predicts that a perfect wave function [$\sigma_M^2(N=1000) = 0$] would have an energy lower than our present wave function by at most 0.18×10^{-5} Ry.

Born-Oppenheimer Approximation

As described in Ref. 32, we have made the Born-Oppenheimer approximation that, from a classical viewpoint, the motion of the electrons is the same as it would be if the nuclei were held fixed in space. This same approximation is stated quantum mechanically by the relation,

$$\Psi(\tilde{X},\tilde{x}) \approx \Psi_N(X)\Psi_e(\tilde{X},\tilde{x}) \qquad (6\text{-}9)$$

where $\Psi_N(\tilde{X})$ and $\Psi_e(\tilde{x})$ are the nuclear and electronic wave functions. In order to test the validity of this assumption, this wave function can be operated on by the exact Hamiltonian:

$$(\frac{-1}{M}\nabla_N^2 - \nabla_e^2 + V_{NN} + V_{Ne} + V_{ee})\Psi_N(\tilde{X})\Psi_e(\tilde{X},\tilde{x}) = E\Psi_N(\tilde{X})\Psi_e(\tilde{X},\tilde{x}), \qquad (6\text{-}10)$$

where ∇_N and ∇_e are the nuclear and electronic gradient operators and M is the mass of the nuclei. After expanding the derivatives Eq. (6-10) becomes

$$\{\frac{-1}{M}\nabla_N\Psi_e\cdot\nabla_N\Psi_N - \frac{1}{M}\Psi_N\nabla_N^2\Psi_e\} + \Psi_e(\frac{-1}{M}\nabla^2\Psi_N)$$

$$+ \Psi_N(-\nabla_e^2\Psi) + (V_{NN} + V_{Ne} + V_{ee})\Psi_N\Psi_e = E\Psi_N\Psi_e . \qquad (6\text{-}11)$$

Since the nuclei are assumed to be fixed in space, the Hamiltonian for the electrons can be defined as

$$H_e = -\nabla_e^2 + V_{NN} + V_{Ne} + V_{ee} \qquad (6\text{-}12)$$

making

$$H = \frac{1}{M}\nabla_N^2 + H_e . \qquad (6\text{-}13)$$

(Many authors separate H such that V_{NN} is not part of H_e.) The function $\Psi_e(\tilde{X},\tilde{x})$, therefore, is defined by the equation

$$H_e\Psi_e(\tilde{X},\tilde{x}) = E_e\Psi(\tilde{X},\tilde{x}) \qquad (6\text{-}14)$$

where E_e is the electronic energy (including V_{NN}). By using Eqs. (6-12) and (6-14) in Eq. (6-11), and neglecting the term in braces, Eq. (6-11) reduces to

$$(\frac{1}{M}\nabla_N^2 + E_e)\Psi_N = E\Psi_N . \qquad (6\text{-}15)$$

This shows that, if the approximation is valid, the electronic energy E_e behaves as the potential energy between the nuclei. This is the Born-Oppenheimer potential calculated here.

It is not necessary, with the methods used here, to make the above approximations; but it is convenient since the corrections are negligible. These corrections have been examined by Laue[33] who finds that the leading term which varies as R_{AB} is of the form of an expectation value of the interaction potential between the atoms divided by the nuclear masses, M. Since this expectation value is of the order of the dip in the Born-Oppenheimer potential, dividing by M makes this correction negligible.

Virial Theorem

Kinetic energy estimates corresponding to the first term in Eq. (6-1) were summed concurrently with Eqs. (2-20) and (2-22). The accuracy of these calculations was enhanced by the differencing technique described in Ch. V and by using the kinetic energy term from Eq. (6-1) rather than from Eq. (2-20) which has singular terms. From these values (Table 6), it is evident that our wave functions do not satisfy the virial theorem. For a di-atomic system in the Born-Oppenheimer approximation, this is[34]

$$R_{AB} \frac{\partial E}{\partial R_{AB}} = -2 \langle T \rangle - \langle V \rangle \qquad (6\text{-}16)$$

where E is the Born-Oppenheimer potential, and <T> and <V> are the values of the kinetic and potential energies from the electronic wave function. By introducing a variational scaling parameter into the wave function ($\Psi(\tilde{x})$ from Tables 4 and 5), and minimizing the energy with respect to this new parameter produces a new wave function,

$$\chi(\tilde{x}) = \Psi(k\tilde{x}) \tag{6-17}$$

which has a lower energy and satisfies the virial theorem. At the potential minimum of $R_{AB} = 5.6\ a_B$, where the term on the left of Eq. (6-16) is zero, the values for the kinetic and total energies from Table 6 were used to calculate a value for k of 0.999977. This gives $\chi(\tilde{x})$ an energy lower than that for $\Psi(\tilde{x})$ by 6.0×10^{-9} Ry.

APPENDIX
THE COMPUTER CODE

The following computer code was used to perform the Monte Carlo sums necessary to determine the energies and kinetic energies from the wave functions already found. The routine used to find the wave functions is very similar to this code but with the following exceptions:

1. The simplex[26] optimization routine is included to minimize the standard deviation of the trial wave functions.
2. A subset of MC points, $\{\tilde{x}\}_{min}$ is reselected from a larger set as in Eq. (3-3).
3. Values of the weight function and the atomic wave function and its derivatives are stored along with these MC point positions.

```
C     MAIN ROUTINE:
C<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
C     Monte Carlo summations corresponding to Eqs.(2-19), (2-20), and (6-1),
C     and kinetic energies from Eq.(6-1) are performed.  Within this loop
C     (statements 4300-12700); electron positions and weight functions are
C     obtained from CONFIG, transformed to the ND internuclear separation
C     distances (DN), then fed into HAM where the wave function and its
C     derivatives are calculated.  At intervals of NTDT iterations, partial
C     sums are printed and random number table (RSEED) is changed.
C<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
      IMPLICIT REAL*8(A-H,O-Z)                                          00000100
      REAL*4 RNDMF                                                      00000200
      COMMON/LLL/DX,DZZ,VOL,RM,REM,HEEC,HMCON,HCON,DNN                  00000300
      COMMON/POL/CP(189),NSSC(189),NTSC(189),NUSC(189)                  00000400
                                                                        00000500
      COMMON/RAN/IXX,IYY                                                00000600
      DIMENSION ELPSM(5,5),EUPSM(5,5),EMPSM(5,5),PSM(5,5),PS(5),EPS(5)  00000700
     X,ESPS(5),DN(5),XP(3,2,4),PAR(41),E(3),ES(5),ANOR(5),PARP(41,5)    00000800
     X ,EAV(5),VAR(5),EDIF(5),SIGDIF(5),X(3,2,4)                        00000900
      DIMENSION ET(5),EPT(5),ESPT(5),ESPF(5),EPF(5),EU(5),PSK(5),       00001000
     XPFK(5),PFKS(5)                                                    00001100
      DATA DN/5.6D0,6.6D0,7.5D0,9.D0,15.D0/                             00001200
      IX=454314877                                                      00001300
      IY=1971698501                                                     00001400
      CALL RSEED(IX,IY)                                                 00001500
      DNN=5.6D0                                                         00001600
      CALL SETUP                                                        00001700
      ND=3                                                              00001800
      NDM=ND-1                                                          00001900
      BD=1.D0                                                           00002000
      READ (5,90000) NSSC,NTSC,NUSC                                     00002100
      READ (5,10000) CP                                                 00002200
      DO 200 J=1,ND                                                     00002300
      DO 100 I=1,ND                                                     00002400
      ELPSM(I,J)=0.D0                                                   00002500
      EUPSM(I,J)=0.D0                                                   00002600
      EMPSM(I,J)=0.D0
```

```
  100 PSM(I,J)=0.D0                                                    00002700
      PS(J)=0.D0                                                       00002800
      EPS(J)=0.D0                                                      00002900
      EPT(J)=0.D0                                                      00003000
      ESPT(J)=0.D0                                                     00003100
      ESPF(J)=0.D0                                                     00003200
      EPF(J)=0.D0                                                      00003300
      PFK(J)=0.D0                                                      00003400
      PSK(J)=0.D0                                                      00003500
      PFKS(J)=0.D0                                                     00003600
      READ (5,10000) (PARP(I,J),I=1,41)                                00003700
      WRITE (6,60000) (PARP(I,J),I=1,41)                               00003800
  200 CONTINUE                                                         00003900
      NT=800                                                           00004000
      NTDT=100                                                         00004100
      NNT=0                                                            00004200
C<<<<<<<<<<<<<<<<<<<<<<<<BEGIN MONTE CARLO LOOP<<<<<<<<<<<<<<<<<<<<<<<<00004300
      DO 1100 I=1,NT                                                   00004400
      NNT=NNT+1                                                        00004500
      CALL CONFIG(X,HINV,NNT)                                          00004600
      DO 800 N=1,ND                                                    00004700
      DO 300 K=1,41                                                    00004800
  300 PAR(K)=PARP(K,N)                                                 00004900
      DO 400 J=1,4                                                     00005000
      DO 400 L=1,2                                                     00005100
      DO 400 K=1,3                                                     00005200
  400 XP(K,L,J)=X(K,L,J)                                               00005300
C<<<<<<<<<<<<<<<<<<<<<<<<TRANSFORM POSITIONS<<<<<<<<<<<<<<<<<<<<<<<<<<00005400
      TRNS=1.D10                                                       00005500
      DO 500 J=1,4                                                     00005600
      ZM=X(3,1,J)-.5D0*DNN                                             00005700
      IF (ZM.LT.-RD) XP(3,2,J)=X(3,1,J)-DN(N)                          00005800
      IF (ZM.GT.RD) XP(3,1,J)=X(3,2,J)+DN(N)                           00005900
      IF (DABS(ZM).GT.RD) GO TO 500                                    00006000
      DZ=1.D0+.5D0*(DN(N)-DNN)/RD                                      00006100
      ZMP=DZ*ZM                                                        00006200
```

```
      TRNS=TRNS*DZ                                                      00006300
      XP(3,1,J)=ZMP+.5D0*DN(N)                                          00006400
      XP(3,2,J)=ZMP-.5D0*DN(N)                                          00006500
  500 CONTINUE                                                          00006600
C<<<<<<<<<<<<<<<<<<<<<<<<<CALL HAM AND BEGIN SUMMATIONS<<<<<<<<<<<<<<<<<00006700
      CALL HAM(XP,EK,E,ANO,PAR,DN(N))                                   00006800
      ES(N)=E(1)                                                        00006900
      ET(N)=E(2)                                                        00007000
      EU(N)=E(3)                                                        00007100
      ANOR(N)=HINV*TRNS*ANO                                             00007200
      IF (ANOR(N).LT.1.D-35) GO TO 700                                  00007300
      EPT(N)=EPT(N)+ET(N)*ANOR(N)                                       00007400
      ESPT(N)=ESPT(N)+ES(N)*ES(N)*ANOR(N)                               00007500
      EPF(N)=EPF(N)+ET(N)*ANOR(N)*ANOR(N)                               00007600
      ESPF(N)=ESPF(N)+ET(N)*ET(N)*ANOR(N)*ANOR(N)                       00007700
      PS(N)=PS(N)+ANOR(N)                                               00007800
      EPS(N)=EPS(N)+ES(N)*ANOR(N)                                       00007900
      PSK(N)=PSK(N)+EU(N)*ANOR(N)                                       00008000
      PFK(N)=PFK(N)+EU(N)*ANOR(N)*ANOR(N)                               00008100
      PFKS(N)=PFKS(N)+EU(N)*EU(N)*ANOR(N)*ANOR(N)                       00008200
      DO 600 M=1,N                                                      00008300
      IF (ANOR(M).LT.1.D-35) GO TO 600                                  00008400
      ANMN=ANOR(M)*ANOR(N)                                              00008500
      ELPSM(M,N)=ELPSM(M,N)+ES(M)*ANMN                                  00008600
      EUPSM(M,N)=EUPSM(M,N)+ES(N)*ANMN                                  00008700
      EMPSM(M,N)=EMPSM(M,N)+ES(M)*ES(N)*ANMN                            00008800
      PSM(M,N)=PSM(M,N)+ANMN                                            00008900
  600 CONTINUE                                                          00009000
  700 CONTINUE                                                          00009100
  800 CONTINUE                                                          00009200
      IF (NNT.NE.NTDT) GO TO 1100                                       00009300
      NNT=0                                                             00009400
      WRITE (6,80000) PS                                                00009500
      WRITE (6,80000) ESPT                                              00009600
      WRITE (6,70000) IXX,IYY                                           00009700
      CALL RSEED(IXX,IYY)                                               00009800
```

```
C<<<<<<<<<<<<<<<<<<<<<<<<FINAL CALCS AND OUTPUT OF VALUES<<<<<<<<<<<<<<<00009900
      DO 900 N=1,ND                                                     00010000
      EFFN=PS(N)*PS(N)/PSM(N,N)                                         00010100
      EAV(N)=EPS(N)/PS(N)                                               00010200
      VAR(N)=EMPSM(N,N)-2.D0*ELPSM(N,N)*EAV(N)+PSM(N,N)*EAV(N)*EAV(N)   00010300
      SIG=DSQRT(VAR(N))/PS(N)                                           00010400
      E2=EPT(N)/PS(N)                                                   00010500
      VAR2=ESPF(N)-2.D0*EPF(N)*E2+PSM(N,N)*E2*E2                        00010600
      SIG2=DSQRT(VAR2)/PS(N)                                            00010700
      EK=PSK(N)/PS(N)                                                   00010800
      VARK=PFKS(N)-2.D0*PFK(N)*EK+PSM(N,N)*EK*EK                        00010900
      SIGK=DSQRT(VARK)/PS(N)                                            00011000
      WRITE (6,20000) DN(N),EAV(N),SIG,EFFN,E2,SIG2,EK,SIGK             00011100
  900 CONTINUE                                                          00011200
      WRITE (6,30000) (DN(J),J=1,NDM)                                   00011300
      DO 1000 N=1,ND                                                    00011400
      DO 1000 M=1,ND                                                    00011500
      IF (M.GE.N) GO TO 1000                                            00011600
      EDIF(M)=EAV(M)-EAV(N)                                             00011700
      VARDIF=VAR(N)*PS(M)*PS(M)+VAR(M)*PS(N)*PS(N)-2.D0*PS(M)*PS(N)*    00011800
     X      (EMPSM(M,N)+EAV(M)*EAV(N)*PSM(M,N)-ELPSM(M,N)*EAV(N)-EUPSM(M,00011900
     X      N)*EAV(M))                                                  00012000
      SIGDIF(M)=DSQRT(VARDIF)/(PS(M)*PS(N))                             00012100
      NM=N-1                                                            00012200
      IF (M.LT.NM) GO TO 1000                                           00012300
      WRITE (6,40000) DN(N),(EDIF(L),L=1,NM)                            00012400
      WRITE (6,50000) (SIGDIF(L),L=1,NM)                                00012500
 1000 CONTINUE                                                          00012600
 1100 CONTINUE                                                          00012700
      STOP                                                              00012800
10000 FORMAT(4E20.10)                                                   00012900
20000 FORMAT(3H E(,F4.1,2H)=,E19.10,3H +-,E14.7,E16.7                   00013000
     X ,2X,E14.7,3H +-,E11.4,E16.7,3H +-,E11.4)                         00013100
30000 FORMAT(/6H    DNN,5(10X,F10.1))                                   00013200
40000 FORMAT(1X,F5.1,5E20.10)                                           00013300
50000 FORMAT(6X,5(5X,E15.6))                                            00013400
```

```
60000  FORMAT(1X,10E13.5)                                        00013500
70000  FORMAT(7H IX,IY=,2I14)                                    00013600
80000  FORMAT(5E20.12)                                           00013700
90000  FORMAT(7(25I3/),14I3)                                     00013800
       END                                                       00013900
```

```
      SUBROUTINE HAM(X,EK,E,PSI2,PAR,DNN)                               00000100
C<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
C     Input parameters are the electron positions (X), the variable
C     parameters from Table 5 (PAR), and the nuclear separation distance
C     (DNN).  Returned values are the square of the wave function (PSI2),
C     the local kinetic energy and local energies corresponding to Eqs.
C     (2-19) and (6-1) (E).
C<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
      IMPLICIT REAL*8 (A-H,O-Z)                                          00000200
      DIMENSION X(3,2,4),R(2,4),VNE(2,4),E(3),XEE(3,4,4),REE(4,4)       00000300
     X,AN(2,4),DDAN(2,4),DAN(3,4,2,4),DA(3,4),DU(3,4),DJAP(3,4)         00000400
     X     ,R2(2,4),RE(2,4)                                             00000500
      COMMON/ALP/ALC(6),ALK(6),EPS,DDD                                  00000600
      COMMON/DIP/DIPC(6),DIPK(6)                                        00000700
      COMMON/QDP/QDC(6),QDK(6)                                          00000800
      COMMON/QQQ/QQC(6),QQK(6)                                          00000900
      COMMON/APA/EEC(3),EEK(3)                                          00001000
      DIMENSION DQU(3,4),PAR(41),ZC(4),ZK(4),DV(3,4),AKNOT(6)           00001100
      DATA AKNOT/2.D0,4.D0,6.D0,7.D0,8.D0,9.D0/                         00001200
      DO 100 I = 1,6                                                    00001300
      ALC(I)=PAR(I)                                                     00001400
      DIPC(I) = PAR(I+6)                                                00001500
      QDC(I)=PAR(I+12)                                                  00001600
      QQC(I)=PAR(I+18)                                                  00001700
      ALK(I)=AKNOT(I)                                                   00001800
      DIPK(I)=AKNOT(I)                                                  00001900
      QDK(I)=AKNOT(I)                                                   00002000
      QQK(I)=AKNOT(I)                                                   00002100
  100 CONTINUE                                                          00002200
      DO 200 I = 1,2                                                    00002300
      EEC(I) = PAR(I+24)                                                00002400
  200 EEK(I) = PAR(I+26)                                                00002500
      EPS=PAR(41)                                                       00002600
      DDD=DNN                                                           00002700
      V = 8.D0/DNN                                                      00002800
C<<<<<<<<<<<<CALCULATE N-E AND E-E POTENTIALS, AND RADIAL DISTANCES<<<<<<<<<<<<<<
```

```
      DO 400 J = 1,4                                                    00003200
      DO 300 I = 1,2                                                    00003300
      R2(I,J)=X(1,I,J)**2+X(2,I,J)**2+X(3,I,J)**2                       00003400
      R(I,J)=DSQRT(R2(I,J))                                             00003500
      RP(I,J)=DSQRT(R2(I,J)+EPS)                                        00003600
      IF (R(I,J).LT..003D0) R(I,J) = .002D0                             00003700
  300 V = V-4.D0/R(I,J)                                                 00003800
  400 CONTINUE                                                          00003900
      DO 700 J = 2,4                                                    00004000
      JM = J-1                                                          00004100
      DO 700 I = 1,JM                                                   00004200
      DO 500 K = 1,3                                                    00004300
      XEE(K,I,J) = X(K,1,J)-X(K,1,I)                                    00004400
  500 XEE(K,J,I) = -XEE(K,I,J)                                          00004500
      REE(I,J) = DSQRT(XEE(1,I,J)**2+XEE(2,I,J)**2+XEE(3,I,J)**2)       00004600
      IF (REE(I,J).LT..003D0) REE(I,J) = .002D0                         00004700
      REE(J,I) = REE(I,J)                                               00004800
      VEE = 2.D0/REE(I,J)                                               00004900
  600 V = V+VEE                                                         00005000
  700 CONTINUE                                                          00005100
      DO 1200 I = 1,2                                                   00005400
      DO 1200 J = 3,4                                                   00005500
      JU = I                                                            00005600
      JD = J                                                            00005700
      CALL POLY(X,R,XEE,REE,NNT,JU,JD,AN(I,J),DJAP,DDAN(I,J))           00005800
      JU=I                                                              00005900
      JD=J                                                              00006000
      DO 800 L = 1,4                                                    00006100
      DO 800 K = 1,3                                                    00006200
  800 DAN(K,L,I,J) = DJAP(K,L)                                          00006300
      RSUM = R(1,I)+R(1,J)+R(2,3-I)+R(2,7-J)                            00006400
      IF (RSUM.GT.20.D0) GO TO 1200                                     00006500
      CALL ALLPOL(X,R,XEE,REE,JU,JD,VV,DV,DDV,RP)                       00006600
      CALL APAREP(XEE,REE,JU,JD,VV,DV,DDV)                              00006700
      IF (DNN.GT.10.D0) GO TO 900                                       00006800
      CALL DIPOLE(X,R,JU,JD,VV,DV,DDV)                                  00006900
```

```
      CALL QUADIP(X,R,JU,JD,VV,DV,DDV,R2)                                00007000
      IF (DNN.GT.7.D0) GO TO 900                                         00007100
      CALL QUAQUA(X,R,JU,JD,VV,DV,DDV,R2)                                00007200
  900 CONTINUE                                                           00007300
      DVDV = 0.D0                                                        00007400
      DADV = 0.D0                                                        00007500
      EX = 0.D0                                                          00007600
      IF (VV.GT.150.) GO TO 1000                                         00007700
      EX = DEXP(-.5D0*VV)                                                00007800
 1000 CONTINUE                                                           00007900
      DO 1100 IE = 1,4                                                   00008000
      DO 1100 K = 1,3                                                    00008100
      DVDV = DVDV+DV(K,IE)*DV(K,IE)                                      00008200
      DADV = DADV+DAN(K,IE,JU,JD)*DV(K,IE)                               00008300
 1100 DAN(K,IE,JU,JD) = EX*(DAN(K,IE,JU,JD)-.5D0*AN(JU,JD)*DV(K,IE))    00008400
      DDAN(JU,JD) = EX*(DDAN(JU,JD)-DADV+AN(JU,JD)*(.25D0*DVDV-.5D0*DDV)00008500
     X     )                                                             00008600
      AN(JU,JD) = EX*AN(JU,JD)                                           00008700
 1200 CONTINUE                                                           00008800
      A = AN(1,3)-AN(2,3)-AN(1,4)+AN(2,4)                                00008900
      IF (DABS(A).LT.1.E-34) A = 1.E-34                                  00009000
      AI = 1.D0/A                                                        00009100
      DO 1300 I = 1,4                                                    00009200
      DO 1300 K = 1,3                                                    00009300
 1300 DA(K,I) = (DAN(K,I,1,3)-DAN(K,I,2,3)-DAN(K,I,1,4)+DAN(K,I,2,4))*AI00009400
      DDA = (DDAN(1,3)-DDAN(2,3)-DDAN(1,4)+DDAN(2,4))*AI                 00009500
C<<<<<<<<<<<<<<<<CALCULATE PSI AND ENERGY<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
      DADA = 0.D0                                                        00009900
      DO 1400 I = 1,4                                                    00010000
      DO 1400 K = 1,3                                                    00010100
 1400 DADA = DA(K,I)*DA(K,I)+DADA                                        00010200
      PSI2 = A*A                                                         00010300
      E(1) = -DDA+V                                                      00010400
      E(2) = DADA+V                                                      00010500
      E(3)=DADA                                                          00010600
      RETURN                                                             00010700
```

```
      SUBROUTINE ALLPOL(X,R,XEE,REE,JU,JD,U,DU,DDU,RP)                  00000100
C<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
C     Input values are the electron positions and radial distances (X, R,
C     XEE, and REE), RP used in Eq. (4-9), and JU and JD which label the
C     permutation from Eq. (4-1).  The function U from Eq. (4-1) and its
C     derivatives DU and DDU are initialized to zero  then contributions
C     from Eq. (4-9) are added.  The parameter EPS, coefficients ALC, and
C     knots ALK are from Table 5.
C<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
      IMPLICIT REAL*8(A-H,O-Z)                                          00000200
      DIMENSION X(3,2,4),R(2,4),DU(3,4),F(2,4),DF(2,4),DDF(2,4),RP(2,4) 00000300
     X,XEE(3,4,4),REE(4,4)                                              00000400
      COMMON/ALP/ALC(6),ALK(6),EPS,DNN                                  00000500
      COMMON/MOMW/MW(2,2)                                               00000600
      MW(1,1) = JU                                                      00000700
      MW(1,2) = JD                                                      00000800
      MW(2,1) = 3-JU                                                    00000900
      MW(2,2) = 7-JD                                                    00001000
      U = 0.D0                                                          00001100
      DDU = 0.D0                                                        00001200
      DO 100 I = 1,4                                                    00001300
      DO 100 K = 1,3                                                    00001400
  100 DU(K,I) = 0.D0                                                    00001500
      DO 200 II = 1,2                                                   00001600
      DO 200 N = 1,2                                                    00001700
      I =MW(N,II)                                                       00001800
      M=3-N                                                             00001900
      CALL SPLINE(6,ALC,ALK,R(N,I),F(N,I),DF(N,I),DDF(N,I),1)           00002000
      DF(N,I)=DF(N,I)/R(N,I)                                            00002100
  200 CONTINUE                                                          00002200
      DO 400 IA = 1,2                                                   00002300
      DO 400 IB = 1,2                                                   00002400
      I =MW(1,IA)                                                       00002500
      J =MW(2,IB)                                                       00002600
      RPE=DSQRT(REE(I,J)**2+EPS)                                        00002700
      FF = F(1,I)*F(2,J)                                                00002800
```

```
      ALL=1.D0/DSQRT(DNN**2+EPS) -1. D0/RP(1,J) -1.D0/RP(2,I) +1. D0/RPE      00002900
      DADF=0.D0                                                               00003000
      DO 300 K=1,3                                                            00003100
      DA1=X(K,2,I)/RP(2,I)**3+XEE(K,I,J)/RPE**3                               00003200
      DA2=X(K,1,J)/RP(1,J)**3-XEE(K,I,J)/RPE**3                               00003300
      DF1=F(2,J)*DF(1,I)*X(K,1,I)                                             00003400
      DF2=F(1,I)*DF(2,J)*X(K,2,J)                                             00003500
      DADF=DADF+DA1*DF1+DA2*DF2                                               00003600
      DU(K,I) = DA1*FF+ALL*DF1       +DU(K,I)                                 00003700
      DU(K,J) = DA2*FF+ALL*DF2       +DU(K,J)                                 00003800
  300 CONTINUE                                                                00003900
      U=U+ALL*FF                                                              00004000
      DDA=6.D0*((REE(I,J)/RPE)**2-1.D0)/RPE**3-3.D0*(((R(2,I)/RP(2,I))**00004100
     X      2-1.D0)/RP(2,I)**3+((R(1,J)/RP(1,J))**2-1.D0)/RP(1,J)**3)        00004200
      DDFF=F(2,J)*(DDF(1,I)+2.D0*DF(1,I))+F(1,I)*(DDF(2,J)+2.D0*DF(2,J))00004300
      DDU=DDU+ DDFF*ALL+2.D0*DADF+FF*DDA                                      00004400
  400 CONTINUE                                                                00004500
      RETURN                                                                  00004600
      END                                                                     00004700
```

```
      SUBROUTINE QUADIP(X,R,JU,JD,U,DU,DDU,R2)                          00000100
C<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
C     Dipole-quadrupole effects, Eq.(4-12), are added to U,DU, and DDU.
C<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
      IMPLICIT REAL*8(A-H,O-Z)                                          00000200
      COMMON/MOMW/MW(2,2)                                               00000300
      DIMENSION X(3,2,4),R(2,4),R2(2,4),DU(3,4),F(2,4),DF(2,4),DDF(2,4) 00000400
      COMMON/QDP/QDC(6),QDK(6)                                          00000500
      DO 100 II = 1,2                                                   00000600
      DO 100 N = 1,2                                                    00000700
      I =MW(N,II)                                                       00000800
      CALL SPLINE(6,QDC,QDK,R(N,I),F(N,I),DF(N,I),DDF(N,I),1)           00000900
      DF(N,I) = DF(N,I)/R(N,I)                                          00001000
  100 CONTINUE                                                          00001100
      DO 300 IA = 1,2                                                   00001200
      DO 300 IB = 1,2                                                   00001300
      I =MW(1,IA)                                                       00001400
      J =MW(2,IB)                                                       00001500
      ZDIF = X(3,1,I)-X(3,2,J)                                          00001600
      ZZ = X(3,1,I)*X(3,2,J)                                            00001700
      CC = 2.D0*(X(1,1,I)*X(1,2,J)+X(2,1,I)*X(2,2,J))-3.D0*ZZ           00001800
      CCP = CC-ZZ                                                       00001900
      DQ = R2(1,I)*X(3,2,J)-R2(2,J)*X(3,1,I)+ZDIF*CC                    00002000
      FF = F(1,I)*F(2,J)                                                00002100
      U = U+DQ*FF                                                       00002200
      DDU = DDU+F(2,J)*(DDF(1,I)*DQ+2.D0*DF(1,I)*(2.D0*DQ+X(3,1,I)*CC+  00002300
     X      X(3,2,J)*R2(1,I)))+F(1,I)*(DDF(2,J)*DQ+2.D0*DF(2,J)*(2.D0*  00002400
     X      DQ-X(3,2,J)*CC-X(3,1,I)*R2(2,J)))                          00002500
      DO 200 K = 1,2                                                    00002600
      DU(K,I) = DU(K,I)+2.D0*(X(3,2,J)*X(K,1,I)+ZDIF*X(K,2,J))*FF+DQ*   00002700
     X      F(2,J)*DF(1,I)*X(K,1,I)                                     00002800
  200 DU(K,J) = DU(K,J)+2.D0*(-X(3,1,I)*X(K,2,J)+ZDIF*X(K,1,I))*FF+DQ*  00002900
     X      F(1,I)*DF(2,J)*X(K,2,J)                                     00003000
      DU(3,I) = DU(3,I)+(CCP+3.D0*X(3,2,J)*X(3,2,J)-R2(2,J))*FF+DQ*F(2, 00003100
     X      J)*DF(1,I)*X(3,1,I)                                         00003200
      DU(3,J) = DU(3,J)+(-CCP-3.D0*X(3,1,I)*X(3,1,I)+R2(1,I))*FF+DQ*F(1,00003300
```

```
   X      I)*DF(2,J)*X(3,2,J)                                    00003400
300 CONTINUE                                                     00003500
    RETURN                                                       00003600
    END                                                          00003700
```

```
      SUBROUTINE QUAQUA(X,R,JU,JD,U,DU,DDU,R2)                          00000100
C<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
C     Quadrupole-quadrupole effects, Eq. (4-13), are added to U, DU, and DDU.
C<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
      IMPLICIT REAL*8(A-H,O-Z)                                          00000200
      COMMON/MCMW/MW(2,2)                                               00000300
      DIMENSION X(3,2,4),R(2,4),R2(2,4),DU(3,4),DA(3,4),F(2,4),DF(2,4), 00000400
     X  DDF(2,4),DQ(3,4)                                                00000500
      COMMON/QCQ/QQC(6),QQK(6)                                          00000600
      DO 100 II=1,2                                                     00000700
      DO 100 N=1,2                                                      00000800
      I=MW(N,II)                                                        00000900
      CALL SPLINE(6,QQC,QQK,R(N,I),F(N,I),DF(N,I),DDF(N,I),1)           00001000
      DF(N,I)=DF(N,I)/R(N,I)                                            00001100
  100 CONTINUE                                                          00001200
      DO 300 IA=1,2                                                     00001300
      DO 300 IB=1,2                                                     00001400
      I=MW(1,IA)                                                        00001500
      J=MW(2,IB)                                                        00001600
      RSUM=R(1,I)*R(1,I)+R(2,J)*R(2,J)                                  00001700
      ZSUM=X(3,1,I)*X(3,1,I)+X(3,2,J)*X(3,2,J)                          00001800
      RR=R(1,I)*R(2,J)                                                  00001900
      ZZ=X(3,1,I)*X(3,2,J)                                              00002000
      CC=X(1,1,I)*X(1,2,J)+X(2,1,I)*X(2,2,J)+ZZ                         00002100
      ZDIF=X(3,1,I)-X(3,2,J)                                            00002200
      ZDIF2=ZDIF*ZDIF                                                   00002300
      QQ=6.D0*CC*(CC-RSUM+5.D0*ZDIF2)+3.D0*RR*RR-15.D0*(X(3,1,I)**2*    00002400
     X      R2(2,J)+X(3,2,J)**2*R2(1,I))+ZZ*(30.D0*RSUM-70.D0*ZSUM+     00002500
     X      105.D0*ZZ)                                                  00002600
      FF=F(1,I)*F(2,J)                                                  00002700
      U=U+QQ*FF                                                         00002800
      DQDF=0.D0                                                         00002900
      DO 200 K=1,2                                                      00003000
      DQ(K,I)=12.D0*CC*(X(K,2,J)-X(K,1,I))+X(K,2,J)*(30.D0*ZDIF2-6.D0*  00003100
     X      RSUM)+X(K,1,I)*(6.D0*R2(2,J)+60.D0*ZZ-30.D0*X(3,2,J)**2)    00003200
      DQ(K,J)=12.D0*CC*(X(K,1,I)-X(K,2,J))+X(K,1,I)*(30.D0*ZDIF2-6.D0*  00003300
```

```
     X      RSUM)+X(K,2,J)*(6.D0*R2(1,I)+60.D0*ZZ-30.D0*X(3,1,I)**2)      00003400
      DU(K,I) = DQ(K,I)*FF+QQ*DF(1,I)*F(2,J)*X(K,1,I)       +DU(K,I)      00003500
      DU(K,J) = DQ(K,J)*FF+QQ*DF(2,J)*F(1,I)*X(K,2,J)       +DU(K,J)      00003600
      DQDF=DQDF+DQ(K,I)*F(2,J)*DF(1,I)*X(K,1,I)+DQ(K,J)*F(1,I)*DF(2,J)*  00003700
     X      X(K,2,J)                                                      00003800
  200 CONTINUE                                                           00003900
      DQ(3,I)=48.D0*CC*ZDIF+X(3,2,J)*(30.D0*ZDIF2+24.D0*RSUM+180.D0*ZZ-  00004000
     X      150.D0*X(3,1,I)**2)-70.D0*X(3,2,J)**3-24.D0*X(3,1,I)*R2(2,J)  00004100
      DQ(3,J)=-48.D0*CC*ZDIF+X(3,1,I)*(30.D0*ZDIF2+24.D0*RSUM+180.D0*ZZ-00004200
     X      150.D0*X(3,2,J)**2)-70.D0*X(3,1,I)**3-24.D0*X(3,2,J)*R2(1,I)  00004300
      DQDF=DQDF+DQ(3,I)*F(2,J)*DF(1,I)*X(3,1,I)+DQ(3,J)*F(1,I)*DF(2,J)*  00004400
     X      X(3,2,J)                                                      00004500
      DU(3,I) = DQ(3,I)*FF+QQ*DF(1,I)*F(2,J)*X(3,1,I)       +DU(3,I)      00004600
      DU(3,J) = DQ(3,J)*FF+QQ*DF(2,J)*F(1,I)*X(3,2,J)       +DU(3,J)      00004700
      DDU= 2.D0*DQDF+QQ*(F(2,J)*(DDF(1,I)+2.D0*DF(1,I))+F(1,I)*(DDF(2,J)  00004800
     X      +2.D0*DF(2,J)))       +DDU                                    00004900
  300 CONTINUE                                                           00005000
      RETURN                                                             00005100
      END                                                                00005200
```

```
      SUBROUTINE APAREP(XEE,REE,JU,JD,U,DU,DDU)                          00000100
C<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
C     The direct electron-electron interactions, Eq.(4-17), are added to
C     U, DU, and DDU.
C<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
      IMPLICIT REAL*8(A-H,O-Z)                                           00000200
      COMMON/MOMW/MW(2,2)                                                00000300
      DIMENSION XEE(3,4,4),REE(4,4),DU(3,4)                              00000400
      COMMON/APA/EEC(3),EEK(3)                                           00000500
      DO 200 IA = 1,2                                                    00000600
      DO 200 IB = 1,2                                                    00000700
      I=MW(1,IA)                                                         00000800
      J=MW(2,IB)                                                         00000900
      IF (REE(I,J).GT.3.D0) GO TO 200                                    00001000
      CALL SPLINE (2,EEC,EEK,REE(I,J),F,DF,DDF,1)                        00001100
      DF = DF/REE(I,J)                                                   00001200
      U = U+F                                                            00001300
      DDU = DDU+2.D0*DDF+4.D0*DF                                         00001400
      DO 100 K = 1,3                                                     00001500
      DU(K,I) = DU(K,I)-DF*XEE(K,I,J)                                    00001600
  100 DU(K,J) = DU(K,J)+DF*XEE(K,I,J)                                    00001700
  200 CONTINUE                                                           00001800
      RETURN                                                             00001900
      END                                                                00002000
```

```
      SUBROUTINE POLY(X,R,XEE,REE,NNT,JU,JD,AJAP,DJAP,DDJAP)            00000100
C<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
C     Values for Schwartz's s-t-u expansion for the atomic wave function
C     (AJAP) and its derivatives (DJAP and DDJAP) are calculated for both
C     atoms.  The values of l, m, and n from Eq. (4-2) are determined by
C     NS, NT, and NU; the coefficients for these terms are given by CP.
C<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
      IMPLICIT REAL*8(A-H,O-Z)                                           00000200
      DIMENSION R(2,4),X(3,2,4),XEE(3,4,4),REE(4,4),DB(2),DDB(2),       00000300
     XDS(2),DT(2),DU(2),DP(3,4),B(2),DJAP(3,4)                          00000400
      DIMENSION SP(18),UP(9),TP(5)                                      00000500
      COMMON/PCL/CP(189),NS(189),NT(189),NU(189)                        00000600
      AK = 3.5D0                                                        00000700
      DDJAP = 0.D0                                                      00000800
      AJAP = 0.D0                                                       00000900
      DO 100 I = 1,4                                                    00001000
      DO 100 K = 1,3                                                    00001100
  100 DJAP(K,I) = 0.D0                                                  00001200
      RSUM = R(1,JU)+R(1,JD)+R(2,3-JU)+R(2,7-JD)                        00001300
      IF (RSUM.GT.20.D0) GO TO 900                                      00001400
      DUDP = 0.D0                                                       00001500
      DDU = 0.D0                                                        00001600
      AK = 3.5D0                                                        00001700
      EX = DEXP(-AK*RSUM*.5D0)                                          00001800
      DO 700 NN = 1,2                                                   00001900
      IF (NN.EQ.1) GO TO 200                                            00002000
      JU = 3-JU                                                         00002100
      JD = 7-JD                                                         00002200
  200 CONTINUE                                                          00002300
      B(NN) = 1.D0                                                      00002400
      DDB(NN) = 0.D0                                                    00002500
      DS(NN) = 0.D0                                                     00002600
      DT(NN) = 0.D0                                                     00002700
      DU(NN) = 0.D0                                                     00002800
      U = REE(JU,JD)                                                    00002900
      S = R(NN,JU)+R(NN,JD)                                             00003000
```

```
      SRS = DSQRT(S)                                                    00003100
      T = R(NN,JU)-R(NN,JD)                                             00003200
      IF (DABS(T).LT.1.D-25)  T = 1.D-25                                00003300
      SI = 1.D0/S                                                       00003400
      TI = 1.D0/T                                                       00003500
      UI = 1.D0/U                                                       00003600
      RUPI = 1.D0/R(NN,JU)                                              00003700
      RDNI = 1.D0/R(NN,JD)                                              00003800
      SI2 = SI*SI                                                       00003900
      T2 = T*T                                                          00004000
      TI2 = TI*TI                                                       00004100
      UI2 = UI*UI                                                       00004200
      ZIT1 = (R(NN,JU)*R(NN,JU)-R(NN,JD)*R(NN,JD)+U*U)*RUPI*UI          00004300
      ZIT2 = (R(NN,JD)*R(NN,JD)-R(NN,JU)*R(NN,JU)+U*U)*RDNI*UI          00004400
      SP(1) = 1.D0                                                      00004500
      UP(1) = 1.D0                                                      00004600
      TP(1) = 1.D0                                                      00004700
      DO 300 I = 2,18                                                   00004800
  300 SP(I) = SP(I-1)*SRS                                               00004900
      DO 400 I = 2,9                                                    00005000
  400 UP(I) = UP(I-1)*U                                                 00005100
      TP(2) = T2                                                        00005200
      TP(3) = T2*T2                                                     00005300
      TP(4) = TP(3)*T2                                                  00005400
      TP(5) = TP(4)*T2                                                  00005500
      D1 = SI*(RUPI+RDNI)                                               00005600
      D2 = TI*(RUPI-RDNI)                                               00005700
      D3 = SI*UI*(ZIT1+ZIT2)                                            00005800
      D4 = TI*UI*(ZIT1-ZIT2)                                            00005900
      DO 500 NPOLY = 2,189                                              00006000
      IP = NS(NPOLY)                                                    00006100
      PI = DFLCAT(IP)*.5D0                                              00006200
      JP = NT(NPOLY)                                                    00006300
      KP = NU(NPOLY)                                                    00006400
      ROOT = SP(IP+1)*TP(JP/2+1)*UP(KP+1)*CP(NPOLY)                     00006500
      B(NN) = B(NN)+ROOT                                                00006600
```

```
      DDB(NN) = DDB(NN)+ROOT*(2.D0*(PI*(PI-1.D0)*SI2+DFLOAT(JP*(JP-1))*  00006700
     X      TI2+DFLOAT(KP*(KP-1))*UI2+PI*D1+DFLOAT(JP)*D2+2.D0*DFLOAT(KP) 00006800
     X      *UI2)+DFLOAT(KP)*PI*D3+DFLOAT(JP*KP)*D4)                      00006900
      DS(NN) = DS(NN)+PI*SI*ROOT                                          00007000
      DT(NN) = DT(NN)+DFLOAT(JP)*TI*RCOT                                  00007100
      DU(NN) = DU(NN)+DFLOAT(KP)*UI2*ROOT                                 00007200
  500 CONTINUE                                                            00007300
      IF (DABS(B(NN)).LT.1.E-35) B(NN) = 1.E-35                           00007400
      DO 600 KX = 1,3                                                     00007500
      DP(KX,JU) = ((DS(NN)+DT(NN))*X(KX,NN,JU)*RUPI-DU(NN)*XEE(KX,JU,JD) 00007600
     X      )/B(NN)                                                       00007700
      DP(KX,JD) = ((DS(NN)-DT(NN))*X(KX,NN,JD)*RDNI+DU(NN)*XEE(KX,JU,JD) 00007800
     X      )/B(NN)                                                       00007900
      DUDP = DUDP+DP(KX,JU)*X(KX,NN,JU)*RUPI+DP(KX,JD)*X(KX,NN,JD)*RDNI   00008000
  600 CONTINUE                                                            00008100
      DDU=DDU+2.D0*(RUPI+RDNI)                                            00008200
  700 CONTINUE                                                            00008300
      DUDU = 4.D0                                                         00008400
      P = B(1)*B(2)                                                       00008500
      DDP = DDB(1)/B(1)+DDB(2)/B(2)                                       00008600
      AJAP = EX*P                                                         00008700
      DDJAP = AJAP*(DDP+AK*(-DUDP-.5D0*DDU+.25D0*AK*DUDU))                00008800
      DO 800 NN = 1,2                                                     00008900
      JU = 3-JU                                                           00009000
      JD = 7-JD                                                           00009100
      DO 800 KX = 1,3                                                     00009200
      DJAP(KX,JD) = AJAP*(-.5D0*X(KX,NN,JD)/R(NN,JD)*AK+DP(KX,JD))        00009300
  800 DJAP(KX,JU) = AJAP*(-.5D0*X(KX,NN,JU)/R(NN,JU)*AK+DP(KX,JU))        00009400
  900 CONTINUE                                                            00009500
      RETURN                                                              00009600
      END                                                                 00009700
```

```
      SUBROUTINE SPLINE(NK,C,AK,X,F,DF,DDF,NS)                          00000100
C<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
C     Spline functions (F) for Eq.(4-14) and its derivatives (DF and DDF)
C     are calculated.  The coefficients and knot positions are given by
C     C and AK.
C<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
      IMPLICIT REAL*8(A-H,O-Z)                                           00000200
      DIMENSION C(NK),AK(NK)                                            00000300
      F = 0.D0                                                          00000400
      DF = 0.D0                                                         00000500
      DDF = 0.D0                                                        00000600
      DO 100 I = 1,NK                                                   00000700
      IF (X.GT.AK(I).AND.NS.EQ.1) GO TO 100                             00000800
      IF (X.LT.AK(I).AND.NS.EQ.-1) GO TO 100                            00000900
      RM = X-AK(I)                                                      00001000
      DDB = C(I)*RM                                                     00001100
      DB = DDB*RM                                                       00001200
      F = F+DB*RM                                                       00001300
      DF = DF+DB                                                        00001400
      DDF = DDF+DDB                                                     00001500
  100 CONTINUE                                                          00001600
      DF = 3.D0*DF                                                      00001700
      DDF = 6.D0*DDF                                                    00001800
      RETURN                                                            00001900
      END                                                               00002000
```

```
      SUBROUTINE SETUP                                                  00000100
C<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
C     This routine generates values for X, PB, P, and XI (Table 1) used to
C     select electron positions with respect to the nuclei.  The Hartree-
C     Fock parameters (CHF and EHF) are found in Ref. 35.  Some values from
C     Table 2 are also defined.
C<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
      IMPLICIT REAL*8 (A-H,O-Z)                                         00000200
      COMMON/LLL/DX,DZ,VOL,RM,REM,HFEC,HMCON,HCON,DNN                   00000300
      DIMENSION PB(65),EHF(5),CHF(5)                                    00000400
      COMMON/AUX/X(65),XI(65),P(65),DRPI(65)                            00000500
      DIMENSION C(3)                                                    00000600
      DATA CHF/.785D0,.2028D0,-.03693D0,-.00293D0,.00325D0/             00000700
      DATA EHF/1.43D0,2.441D0,4.10D0,6.484D0,.7978D0/                   00000800
      DO 100 I = 1,5                                                    00000900
  100 CHF(I) = DSQRT((EHF(I)*2.D0)**3/2.D0)*.282095*CHF(I)              00001000
      WRITE (6,10000)                                                   00001100
      X(1) = 0.D0                                                       00001200
      XI(1) = 0.D0                                                      00001300
      DO 300 I = 2,65                                                   00001700
      Q=.15D0*DFLOAT(I-1)                                               00001800
      PB(I) = 0.D0                                                      00001900
      DO 200 J = 1,5                                                    00002000
  200 PB(I) = PB(I)+CHF(J)*DEXP(-Q*EHF(J))                              00002100
      PB(I) = PB(I)**2                                                  00002200
      P(I) = PB(I)*Q**2                                                 00002300
      P(I)=DMAX1(P(I),.001D0)                                           00002400
      C(1)=3.D0                                                         00002500
      C(2)=.5D0                                                         00002600
      C(3)=3.5D0                                                        00002700
      C111=C(1)*DSQRT(C(2))                                             00002800
      QM=Q-C(3)                                                         00002900
      IF (QM.GT.0.D0) QM=DSQRT(QM)                                      00003000
      P(I)=P(I)*(1.D0+C111*DEXP(-C(2)*QM*QM))                           00003100
  300 CONTINUE                                                          00003200
      P(2) = P(3)                                                       00003300
```

```
      P(1) = P(2)                                                        00003400
      DO 400 I = 2,65                                                    00003500
      IM = I-1                                                           00003600
      X(I)=.15D0*DFLOAT(I-1)                                             00003700
      XI(I)=.075D0*(P(I)+P(IM))+XI(IM)                                   00003800
      DRPI(IM)=(P(I)-P(IM))/.15D0                                        00003900
      WRITE (6,20000) X(I),PB(I),P(I),XI(I)                              00004000
  400 CONTINUE                                                           00004100
      DZ=60.D0                                                           00004200
      DX=40.D0                                                           00004300
      VOL=DX*DX*DZ                                                       00004400
      RM=.5D0*DNN                                                        00004500
      REM=1.3D0                                                          00004600
      HEEC=XI(65)/REM                                                    00004700
      HMCON=16.D0*XI(65)/DNN**2                                          00004800
      HCON=XI(65)*12.566370620/VOL                                       00004900
      RETURN                                                             00005000
10000 FORMAT(8H  WT FUN,11X,1HX,18X,2HPB,19X,1HP,18X,2HXI)               00005100
20000 FORMAT(4E20.6)                                                     00005200
      END                                                                00005300
```

```
      SUBROUTINE CONFIG(X,WTAV,NNT)                                          00000100
C<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
C     Electron positions (X), one-electron weight functions (H), and the
C     final averaged weight function from Eq.(2-32) (1/WTAV) are determined.
C<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
      IMPLICIT REAL*8 (A-H,O-Z)                                              00000200
      REAL*4 RNDMF                                                           00000300
      COMMON/LLL/DX,DZ,VOL,RM,REM,HEEC,HMCON,HCON,DNN                        00000400
      DIMENSION X(3,2,4),H(8,4),HAV(4),CON(4)                                00000500
      HEC = 7.D0                                                             00000600
      DO 800 I = 1,4                                                         00000700
      NCN = 5                                                                00000800
      IF (I.EQ.1.OR.I.EQ.3) NCN = 85                                         00000900
      II = I                                                                 00001000
C<<<<<<<<<<<<<<<<<<<<<<<<<<<<<PICK POSITIONS<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
      TP = 100.D0+HEC*DFLOAT(I-1)                                            00001400
      IF (I.EQ.1) TP = 110.D0                                                00001500
  100 CONTINUE                                                               00001600
      IW = RNDMF(1.)*TP                                                      00001700
      IF (IW.LT.90) GO TO 200                                                00001800
      IF (IW.LT.93) GO TO 300                                                00001900
      IF (IW.LT.100) GO TO 400                                               00002000
      IF (I.EQ.1) GO TO 400                                                  00002100
      IE = 1.D0+RNDMF(1.)*DFLOAT(I-1)                                        00002200
      REE = REM*RNDMF(1.)                                                    00002300
      AC = 2.D0*RNDMF(1.)-1.D0                                               00002400
      AS = DSQRT(1.D0-AC*AC)                                                 00002500
      PHI = 6.28318530800*RNDMF(1.)                                          00002600
      X(1,1,I) = X(1,1,IE)+REE*AS*DCOS(PHI)                                  00002700
      X(2,1,I) = X(2,1,IE)+REE*AS*DSIN(PHI)                                  00002800
      X(3,1,I) = X(3,1,IE)+REE*AC                                            00002900
      IF (DABS(X(1,1,I)).GT..5D0*DX) GO TO 100                               00003000
      IF (DABS(X(2,1,I)).GT..5D0*DX) GO TO 100                               00003100
      IF (X(3,1,I).LT.RM-.5D0*DZ.OR.X(3,1,I).GT.RM+.5D0*DZ) GO TO 100        00003200
      GO TO 500                                                              00003300
  200 CONTINUE                                                               00003400
```

```
      CALL XFIND(X,II)                                                  00003500
      IF (IW.GE.NCN) X(3,1,I) = X(3,1,I)+DNN                            00003600
      GO TO 500                                                         00003700
  300 CONTINUE                                                          00003800
      X(1,1,I) = (RNDMF(1.)-.5D0)*DX                                    00003900
      X(2,1,I) = (RNDMF(1.)-.5D0)*DX                                    00004000
      X(3,1,I) = RNDMF(1.)*DZ+.5D0*(DNN-DZ)                             00004100
      GO TO 500                                                         00004200
  400 CONTINUE                                                          00004300
      RAN = RNDMF(1.)                                                   00004400
      RW = RM*DSQRT(RAN)                                                00004500
      AC = 2.D0*RNDMF(1.)-1.D0                                          00004600
      AS = DSQRT(1.D0-AC*AC)                                            00004700
      PHI = 6.283185308D0*RNDMF(1.)                                     00004800
      X(1,1,I) = RW*AS*DCOS(PHI)                                        00004900
      X(2,1,I) = RW*AS*DSIN(PHI)                                        00005000
      X(3,1,I) = RW*AC+RM                                               00005100
  500 CONTINUE                                                          00005200
C<<<<<<<<<<<<<<<<<<<<<<<<CALCULATE H VALUES<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
C<<<<<<<<<<<<<<<<<<HN1=H(1),HN2=H(2),HCON=H(3),HM=H(4),HEE=H(5-8)<<<<<<<<<<<<<<<<<
      X(1,2,I) = X(1,1,I)                                               00005700
      X(2,2,I) = X(2,1,I)                                               00005800
      X(3,2,I) = X(3,1,I)-DNN                                           00005900
      DO 600 K = 4,8                                                    00006000
  600 H(K,I) = 0.D0                                                     00006100
      R1 = DSQRT(X(1,1,I)**2+X(2,1,I)**2+X(3,1,I)**2)                   00006200
      R2 = DSQRT(X(1,2,I)**2+X(2,2,I)**2+X(3,2,I)**2)                   00006300
      CALL FINDH(R1,HN1)                                                00006400
      CALL FINDH(R2,HN2)                                                00006500
      H(1,I) = HN1                                                      00006600
      H(2,I) = HN2                                                      00006700
      RP = DSQRT(4.D0*(X(1,1,I)**2+X(2,1,I)**2)+(X(3,1,I)+X(3,2,I))**2) 00006800
      IF (RP.LT.DNN) H(4,I) = HMCON/RP                                  00006900
      H(3,I) = HCON                                                     00007000
      CON(I) = 3.D0*H(3,I)+7.D0*H(4,I)                                  00007100
      DO 700 IE = 1,I                                                   00007200
```

```
      IF (IE.EQ.I) GO TO 700                                                    00007300
      REE = DSQRT((X(1,1,IE)-X(1,1,I))**2+(X(2,1,IE)-X(2,1,I))**2+(X(3,        00007400
     X      1,IE)-X(3,1,I))**2)                                                00007500
      IF (REE.LT.REM) H(4+IE,I) = HEEC/REE**2                                   00007600
      H(4+I,IE) = H(4+IE,I)                                                     00007700
  700 CONTINUE                                                                  00007800
  800 CONTINUE                                                                  00007900
C<<<<<<<<<<<<<<<<AVERAGE OVER ALL WAYS OF PICKING THE CONFIGURATION<<<<<<<<<<<<<<<
      HAVAV = 0.D0                                                              00008300
      AON = 85.D0                                                               00008400
      DO 900 I1=1,4                                                             00008500
      I2=3-I1                                                                   00008600
      IF (I1.GE.3) I2=7-I1                                                      00008700
      DO 900 IDUM=3,4                                                           00008800
      I3=IDUM                                                                   00008900
      IF (I1.GE.3) I3=IDUM-2                                                    00009000
      I4=10-I1-I2-I3                                                            00009100
      HAV1 = CON(I1)+AON*H(1,I1)+(90.D0-AON)*H(2,I1)                            00009200
      HAV1 = HAV1+10.D0*H(4,I1)                                                 00009300
      HAV2 = CON(I2)+AON*H(2,I2)+(90.D0-AON)*H(1,I2)+HEC*H(4+I1,I2)             00009400
      HAV3 = CON(I3)+AON*H(1,I3)+(90.D0-AON)*H(2,I3)+HEC*(H(4+I1,I3)+           00009500
     X      H(4+I2,I3))                                                        00009600
      HAV4 = CON(I4)+AON*H(2,I4)+(90.D0-AON)*H(1,I4)+HEC*(H(4+I1,I4)+           00009700
     X      H(4+I2,I4)+H(4+I3,I4))                                             00009800
      HAVAV = HAVAV+HAV1*HAV2*HAV3*HAV4                                         00009900
  900 CONTINUE                                                                  00010000
      WTAV = 1.D0/HAVAV                                                         00010100
      RETURN                                                                    00010200
      END                                                                       00010300
```

```
      SUBROUTINE XFIND(Z,I)                                                  00000100
C<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
C     Table generated in SETUP is used to find electron positions picked
C     with respect to a nucleus.
C<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
      IMPLICIT REAL*8 (A-H,O-Z)                                              00000200
      REAL*4 RNDMF                                                          00000300
      DIMENSION Z(3,2,4)                                                    00000400
      COMMON/AUX/X(65),XI(65),P(65),DRPI(65)                                00000500
      Y = RNDMF(1.)*XI(65)                                                  00000600
      CALL LOCATE(Y,XI,J,0,65)                                              00000700
      EP = 2.D0*(Y-XI(J))                                                   00000800
      R = X(J)+EP/(P(J)+DSQRT(P(J)**2+DRPI(J)*EP))                          00000900
      PHI = 6.283185308D0*RNDMF(1.)                                         00001000
      AMU = 2.D0*RNDMF(1.)-1.D0                                             00001100
      AMSU = DSQRT(1.D0-AMU**2)                                             00001200
      Z(1,1,I) = R*AMSU*DSIN(PHI)                                           00001300
      Z(2,1,I) = R*AMSU*DCOS(PHI)                                           00001400
      Z(3,1,I) = R*AMU                                                      00001500
      RETURN                                                                00001600
      END                                                                   00001700
```

```
      SUBROUTINE FINDH(R,H)                                              00000100
C<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
C     Table generated in SETUP is used to determine the 1-electron weight
C     function (H1 or H2) with respect to a nucleus position.
C<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
      IMPLICIT REAL*8 (A-H,O-Z)                                          00000200
      COMMON/AUX/XB(65),XBI(65),P(65),DRPI(65)                           00000300
      IF (R.GE.9.6D0) GO TO 100                                          00000400
      L=R/.15D0+1.D0                                                     00000500
      WI = P(L)+(R-XE(L))*DRPI(L)                                        00000600
      H = WI/R**2                                                        00000700
      GO TO 200                                                          00000800
  100 H = 0.D0                                                           00000900
  200 RETURN                                                             00001000
      END                                                                00001100
```

```
      SUBROUTINE LOCATE(X,R,J,NSIZ,NDIM)                                00000100
C<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
C     This routine is used by XFIND to find the value of m (J) in Eq.(2-6).
C<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
      IMPLICIT REAL*8 (A-H,O-Z)                                         00000200
      DIMENSION R(NDIM),IC(6)                                           00000300
      DATA IC/32,16,8,4,2,1/                                            00000400
      J = IC(1-NSIZ)+1                                                  00000500
      N3 = 2-NSIZ                                                       00000600
      IF (N3.EQ.7) GO TO 300                                            00000700
      DO 200 IL = N3,6                                                  00000800
      IF (X.GT.R(J)) GO TO 100                                          00000900
      J = J-IC(IL)                                                      00001000
      GO TO 200                                                         00001100
  100 J = J+IC(IL)                                                      00001200
  200 CONTINUE                                                          00001300
  300 CONTINUE                                                          00001400
      IF (X.GT.R(J)) GO TO 400                                          00001500
      J = J-1                                                           00001600
  400 RETURN                                                            00001700
      END                                                               00001800
```

```
      FUNCTION RNDMF(X)                                                 00000100
C<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
C     This routine selects randomly a value (RNDMF) from a random number table
C     then replaces it with a new random number.  This random number generator
C     is explained in Ref. 36.
C<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
      COMMON/RAN/IX,IY,ITAB(128)                                        00000200
      IY = IY*65539                                                     00000300
      IF (IY) 100,200,200                                               00000400
  100 IY = IY+2147483647+1                                              00000500
  200 IT = IY*.591389E-07+1.                                            00000600
C<<<<<<IBM GIVES THIS CONSTANT AS .4656613E-9 BUT I HAVE DROPPED THE 3 TC<<<<<<<
C<<<<<<<<<<<<<<<<<<<<<<<<AVOID  EVER GETTING A NUMBER EQUAL TO 1<<<<<<<<<<<<<<<<<<
      RNDMF = ITAB(IT)*.465661E-9                                       00001100
      IX = IX*65549                                                     00001200
      IF (IX) 300,400,400                                               00001300
  300 IX = IX+2147483647+1                                              00001400
  400 ITAB(IT) = IX                                                     00001500
      RETURN                                                            00001600
      END                                                               00001700
```

```
      SUBROUTINE RSEED(IA,IB)
C<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
C     Random number table (ITAB) is generated for use by RNDMF.
C<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
      COMMON/RAN/IX,IY,ITAB(128)                                          00000200
      IX = IA                                                             00000300
      IY = IB                                                             00000400
      DO 300 I = 1,128                                                    00000500
      IX = IX*65549                                                       00000600
      IF (IX) 100,200,200                                                 00000700
  100 IX = IX+2147483647+1                                                00000800
  200 ITAB(I) = IX                                                        00000900
  300 CONTINUE                                                            00001000
      RETURN                                                              00001100
      END                                                                 00001200
```

REFERENCES

1. L. W. Bruch and I. J. McGee, J. Chem. Phys. 52, 5884 (1970).

2. H. G. Bennewitz, H. Busse, and H. D. Dohmann, Chem. Phys. Lett. 8, 235 (1970).

3. J. M. Farrar and Y. T. Lee, J. Chem. Phys. 56, 5801 (1972.

4. A. L. J. Burgmans, J. M. Farrar, and Y. T. Lee, J. Chem. Phys. 64, 1345 (1976).

5. R. Chapman, Phys. Rev. A 12, 2333 (1975).

6. J. C. Slater and J. G. Kirkwood, Phys. Rev. 37, 682 (1931).

7. H. Margenau, Phys. Rev. 38, 747 (1931).

8. D. R. McLaughlin and H. F. Schäfer III, Chem. Phys. Lett. 12, 244 (1971).

9. P. Bertoncini and A. C. Wahl, Phys. Rev. Lett. 25, 991 (1970).

10. B. Liu and A. D. McLean, J. Chem. Phys. 59, 4557 (1973).

11. P. Bertoncini and A. C. Wahl, J. Chem. Phys. 58, 1259 (1973).

12. P. D. Dacre, Chem. Phys. Lett. 50, 147 (1977).

13. P. G. Burton, J. Chem. Phys. 70, 3112 (1979).

14. G. Chalasinski and B. Jeziorski, Mol. Phys. 32, 81 (1976).

15. R. L. Coldwell, Int. J. of Quantum Chem. 11S, 215 (1977).

16. R. L. Coldwell and R. E. Lowther, Int. J. of Quantum Chem. 12S, 329 (1978).

17. R. L. Coldwell, Int. J. of Quantum Chem. 13S, 705 (1979).

18. N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, J. Chem. Phys. 21, 1087 (1953).

19. W. L. McMillan, Phys. Rev. A 138, 442 (1965).

20. J. W. Moskowitz and M. H. Kalos, submitted for publication in J. Chem. Phys.

21. J. A. Pople, R. Seeger, R. Krishnan, Int. J. of Quantum Chem. 11S, 149 (1977).

22. E. A. Hylleraas, Z. Physik 54, 347 (1929).

23. T. Kinoshita, Phys. Rev. 115, 366 (1959).

24. C. L. Pekeris, Phys. Rev. 115, 1216 (1959).

25. C. Schwartz, Phys. Rev. 128, 1146 (1962).

26. J. Kolwalik and M. R. Osborne, Methods for Unconstrained Optimization Problems (American Elsevier, New York, 1968).

27. J. M. Hammersley and D. C. Handscomb, Monte Carlo Methods (Methuen and Co., London, 1964).

28. A. A. Frost, R. E. Kellog, and E. C. Curtis, Rev. Mod. Phys. 32, 313 (1960).

29. R. L. Coldwell, unpublished.

30. G. Starkschall and R. G. Gordon, J. Chem. Phys. 54 663 (1971).

31. H. M. James and A. S. Coolidge, J. Chem. Phys. 1, 825 (1933).

32. H. Eyring, J. Walter, and G. E. Kimball, Quantum Chemistry (John Wiley and Sons Inc., London, 1946).

33. H. Laue, J. Chem. Phys. 46, 3034 (1967).

34. B. L. Moisiewitsch, Variational Principles (John Wiley and Sons Ltd., London, 1966).

35. E. Clementi, Tables of Atomic Functions (IBM Corp., San Jose, 1965).

36. R. L. Coldwell, J. of Computational Phys. 14, 223 (1974).

BIOGRAPHICAL SKETCH

Rex Everett Lowther was born in Syracuse, New York, on December 25, 1951, as the first son of Mr. and Mrs. John Martin Lowther. After being enrolled at the University of Florida for one year, he attended Alfred University for three years earning a baccalaureate in physics in May 1974. He was admitted as a graduate student in the Physics Department, University of Florida, in September 1974.

He was married to the former Joni Lynn Walker on May 10, 1980.

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Arthur A. Broyles, Chairman
Professor of Physics

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Charles A. Burnap
Assistant Professor of
Mathematics

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Jim R. Brookeman
Associate Professor of Physics

This dissertation was submitted to the Graduate Faculty of the Department of Physics in the College of Liberal Arts and Sciences and to the Graduate Council, and was accepted as partial fulfillment of the requirements for the degree of Doctor of Philosophy.

December, 1980

Dean for Graduate Studies and
Research